Table of Contents

Model Documentation

Model Detail

This document provides a complete overview of all element details. For simpler and more focused reports, simply copy this initial template and turn off the sections not required.

Design Model

Type: **Package**
Status: Proposed. Version . Phase 1.0.
Package: Model
Detail: *Created on 2.10.2011. Last modified on 2.10.2011*
GUID: {033FB2F9-7370-4078-BC5F-7D19976BDC9F}

BL

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Design Model
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {D3D273E2-87B2-414f-97DE-BF381B0690A8}

Ukol

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: BL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {DAF70164-820E-4faf-8B00-CC9F8933F92F}

Ukol - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
GUID: {3F051BE2-346F-4610-ACBC-F5BAB489FE2E}
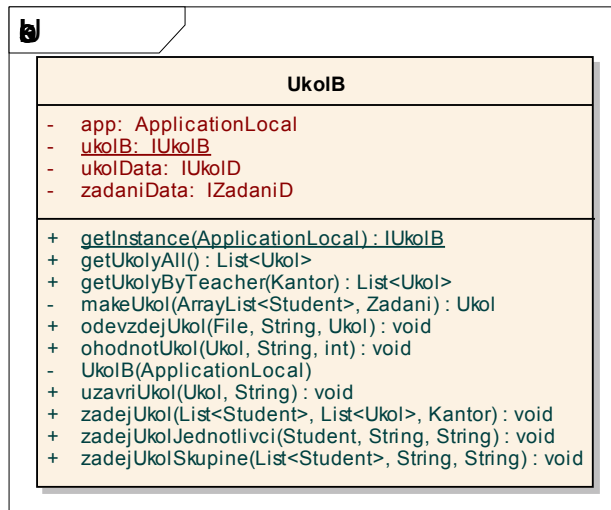
Figure: 1

UkolB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Ukol *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{10A2B788-7F07-4bcf-BE7C-C85EAF46221B}

Singleton trida, ktera implementuje chovani IUkolB a zprostredkovava sluzby business logiky.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public UkolB	Private ukolData IUkolD	
Association Source -> Destination	Public UkolB	Private zadaniData IZadaniD	
Realization Source -> Destination	Public UkolB	Public IUkolB	
Association Source -> Destination	Public AssignTask	Package taskB UkolB	
Association Source -> Destination	Public UkolB	Private app ApplicationLocal	
Association Source -> Destination	Public UkolB	Private ukolB IUkolB	

Association Source -> Destination	Public studentTask	Package ukolB UkolB	

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
ukolB IUkolB Private Static		*Default:*
ukolData IUkolD Private		*Default:*
zadaniData IZadaniD Private		*Default:*

Method	Notes	Parameters
Static **getInstance()** IUkolB Public		**ApplicationLocal** [in] **app**
getUkolyAll() List<Ukol> Public	Metoda pozada datovou vrstu o vsechny ukoly v systemu a vrati je v datove strukture List @return List vsech ukolu v databazi	
getUkolyByTeacher() List<Ukol> Public	Metoda pozada datovou vrstu o vsechny ukoly daneho ucitele a vrati jejich list @return List ukolu danneho ucitele.	**Kantor** [in] **teacher**

makeUkol() Ukol Private		**ArrayList<Student>** [in] **vypracovateleUkolu** **Zadani** [in] **temp**
odevzdejUkol() void Public	Metoda zpracuje vstupni soubor priradiho k uzivateli a k ukolu ktereho se tyka a takto upravenou strukturu preda do datove vrstvy k ulozeni	**File** [in] **soubor** **String** [in] **loginUzivatele** **Ukol** [in] **ukolOdevzdani** ukolOdevzdani
ohodnotUkol() void Public	Metoda prida bodove a slovni hodnoceni, nebo jen jedno z nich na zǎˇkladǍ› vstupnǍch parametrĹŻ ke konkrǍ©tnǍ©mu Ǎşkolu. Pote pozada datovou vrstvu o vlozeni upravene struktury do databaze. @param ukol	**Ukol** [in] **u** **String** [in] **slovniHodnoceni** **int** [in] **bodyHodnoceni** bodyHodnoceni
UkolB() Private		**ApplicationLocal** [in] **app**
uzavriUkol() void Public	Metoda zmeni stav ukolu na stav Ukonceno. Pote pozada datovou vrstvu o zapis upravene struktury do DB	**Ukol** [in] **ukol** ukol **String** [in] **stav**
zadejUkol() void Public	Metoda zada mnozinu ukolu, mnozine studentu, ktere vyucuje urcity kantor. Pote pozada datovou vrstu o vlozeni teto struktury do DB.	**List<Student>** [in] **students** **List<Ukol>** [in] **tasks** tasks **Kantor** [in] **loggedKantor**
zadejUkolJednotlivci() void Public	Metoda priradi studentovy novy ukol a pote pozada datovou vrstu o zapis upravenych dat do DB	**Student** [in] **vypracovatelUkolu** vypracovatelUkolu **String** [in] **nazevZadani** **String** [in] **nazevUkolu**
zadejUkolSkupine() void Public	Metoda priradi ukol skupine studentu a pote zavola datovou vrstu a pozada ji o vlozeni novych udaju	**List<Student>** [in] **vypracovateleUkolu** **String** [in] **nazevZadani** nazevZadani **String** [in] **nazevUkolu**

Uzivatel

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	BL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{9FCD90C2-7BDB-48fd-9CEF-C72C23274E54}

Uzivatel - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{7CAAA1CB-3858-465a-A3A6-026A16852D90}

UzivatelB

- app: ApplicationLocal
- userData: IUzivatelD
- uzivatelB: IUzivatelB

+ addUzivatel(String, String, String, String, String, String) : void
+ addUzivatel(InputStream) : void
+ findUserByLogin(String) : Uzivatel
+ findUserToLogin(String) : Uzivatel
+ getAllKantor() : List<Kantor>
+ getInstance(ApplicationLocal) : IUzivatelB
+ getKantor(Zadani) : void
+ getStudenstByKantor(Kantor) : List<Student>
+ getStudentyAll() : List<Student>
+ getUzivateleAll() : List<Uzivatel>
+ modifyUser(Uzivatel) : void
+ removeUser(String) : void
+ resetPassword(Uzivatel) : void
+ updateUdaje(String, String, String) : void
- UzivatelB(ApplicationLocal)

Figure: 2

UzivatelB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Uzivatel *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{AA1A8F47-D5D3-4a0b-B4DA-2A06EE7023E1}

Singleton trida, ktera implementuje chovani IUzivatelB a zprostredkovava sluzby business logiky.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private app	

Source -> Destination	UzivatelB	ApplicationLocal	
Association Source -> Destination	Public UzivatelB	Private userData IUzivatelD	
Association Source -> Destination	Public UzivatelB	Private uzivatelB IUzivatelB	
Association Source -> Destination	Public AssignTask	Package userB UzivatelB	
Realization Source -> Destination	Public UzivatelB	Public IUzivatelB	
Association Source -> Destination	Public studentTask	Package userB UzivatelB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
userData IUzivatelD Private		*Default:*
uzivatelB IUzivatelB Private Static		*Default:*

Operations

Method	Notes	Parameters
addUzivatel() void Public	metoda pro vkladani studentu a kantoru pokud je cokoli nulove tak hazi EJBException pokud je login a email ne-unikatni tak hazi EJBException	**String** [in] **login** **String** [in] **heslo** **String** [in] **jmeno**

		String [in] **prijmeni**
		String [in] **mail**
		String [in] **status** - zda se jedna o studenta ci kantora, pokud "student" tak je to student, cokoli jineho je kantor
addUzivatel() void Public	Metoda vytvori uzivatele, ktere jsou ve vstupnim souboru. OÄŤekÄˇvanÄ˝ formÄˇt je CSV, potÄ© poĹľÄˇdÄˇ datovou vrstu o zÄˇpis tÄ›chto uĹľivatelĹˇ do DB	InputStream [in] **f** f
findUserByLogin() Uzivatel Public	Metoda poĹľÄˇdÄˇ datovou vrstu o vyhledÄˇnÄ uĹľivatele na zÄˇkladÄ› jeho loginu. @return Uzivatele nalezenÄ©ho na zÄˇkladÄ› loginu.	String [in] **login**
findUserToLogin() Uzivatel Public		String [in] **login**
getAllKantor() List<Kantor> Public	Metoda pozada datovou vrstu o vsechny ucitele v DB @return List vsech ucitele v DB	
Static **getInstance()** IUzivatelB Public		ApplicationLocal [in] **app**
getKantor() void Public	Metoda pozada datovou vrstu o kantora, ktery vytvoril specificke zadani	Zadani [in] **z** z
getStudenstByKantor() List<Student> Public	Metoda poĹľÄˇdÄˇ datovou vrstu o vyhledÄˇnÄ vĹˇech studentĹˇ, kterÄ© uÄŤÄ kantor. @return List studentĹˇ, vyhledanÄ˝ch podle kantora.	Kantor [in] **teacher**
getStudentyAll() List<Student> Public	Metoda pozada datovou vrstu o vsechny studenty v DB. @return List studentu.	
getUzivateleAll() List<Uzivatel> Public	Metoda pozada datovou vrstu o vsechny uzivatele v DB @return List vsech uzivatelu v DB	
modifyUser() void Public	Metoda pozada datovou vrstu o upraveni aktualniho studenta, podle jeho loginu, ktery se nemeni.	Uzivatel [in] **user** user
removeUser() void Public	Metoda poĹľÄˇdÄˇ datovou vrstvu o odstranÄ›nÄ z databÄˇze uĹľivatele podle jeho loginu.	String [in] **login** login
resetPassword() void	Metoda nastavi uzivateli vychozi heslo.	Uzivatel [in] **user**

Public		
updateUdaje() void Public	Metoda upravi uzivateli na zaklade jeho loginu, jeho e-mail a heslo. Pote pozada datovou vrstu o vlozeni takto upravenych udajou do DB	**String** [in] **login** **String** [in] **mail** **String** [in] **heslo**
UzivatelB() Private		**ApplicationLocal** [in] **app**

IBussiness

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: BL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {67AA44F5-0561-4b3b-8659-CB68CDDC58BB}

IBussiness - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
GUID: {F29217B8-F9FC-47e2-8571-563CD2097EB1}
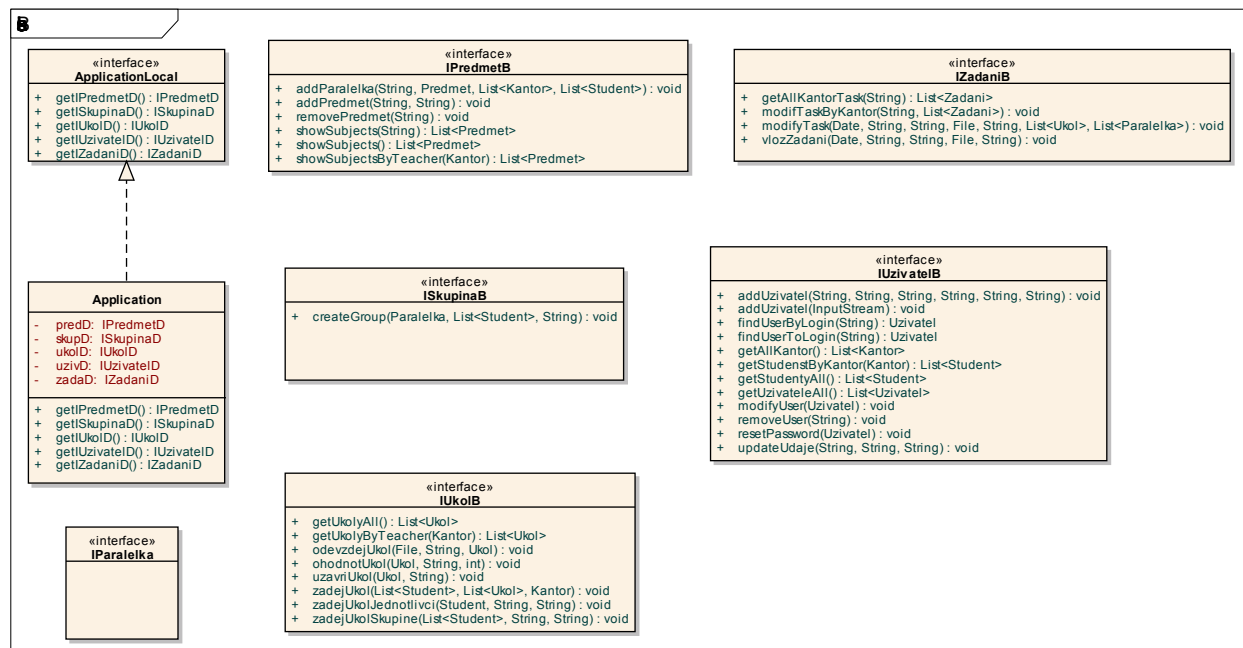
Figure: 3

Application

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: IBussiness   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {CA78EAF7-CFA0-4c92-8211-B1620DE51235}

Fascada k pristupu do datove vrstvy, vyuziva dependency injection.

Custom Properties

- isActive = False

Tagged Values

- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public Application	Public ApplicationLocal	
Association Source -> Destination	Public Application	Private uzivD IUzivatelD	
Association Source -> Destination	Public Application	Private predD IPredmetD	
Association Source -> Destination	Public Application	Private ukolD IUkolD	
Association Source -> Destination	Public Application	Private zadaD IZadaniD	
Association Source -> Destination	Public Application	Private skupD ISkupinaD	

Attributes

Attribute	Notes	Constraints and tags
predD IPredmetD Private		*Default:* [annotations = @Inject]

skupD ISkupinaD Private		Default: [annotations = @Inject]
ukolD IUkolD Private		Default: [annotations = @Inject]
uzivD IUzivatelD Private		Default: [annotations = @Inject]
zadaD IZadaniD Private		Default: [annotations = @Inject]

Operations

Method	Notes	Parameters
getIPredmetD() IPredmetD Public		
getISkupinaD() ISkupinaD Public		
getIUkolD() IUkolD Public		
getIUzivatelD() IUzivatelD Public	Add business logic below. (Right-click in editor and choose "Insert Code > Add Business Method")	
getIZadaniD() IZadaniD Public		

ApplicationLocal

Type:	**Interface**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	IBussiness *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{A5776133-CAAC-409b-B2F3-B3B27293F312}

Interface k fasade, ktera zajistuje pristup do datove vrstvy.

Tagged Values
- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AssignTask	Package app ApplicationLocal	
Association Source -> Destination	Public PredmetB	Private app ApplicationLocal	
Association Source -> Destination	Public ParalelkaB	Private app ApplicationLocal	
Association Source -> Destination	Public ZadaniB	Private app ApplicationLocal	
Realization Source -> Destination	Public Application	Public ApplicationLocal	
Association Source -> Destination	Public FileBean	Package app ApplicationLocal	
Association Source -> Destination	Public GroupBean	Package app ApplicationLocal	
Association Source -> Destination	Public taskBean	Package app ApplicationLocal	
Association Source -> Destination	Public UzivatelB	Private app ApplicationLocal	
Association Source -> Destination	Public CheckStudentBean	Package app ApplicationLocal	
Association Source -> Destination	Public testBean	Package app ApplicationLocal	
Association Source -> Destination	Public LoginBean	Package app ApplicationLocal	
Association	Public	Package app	

Source -> Destination	SubjectBean	ApplicationLocal	
Association Source -> Destination	Public UserBean	Package app ApplicationLocal	
Association Source -> Destination	Public ParalelkaBean	Package app ApplicationLocal	
Association Source -> Destination	Public UkolB	Private app ApplicationLocal	
Association Source -> Destination	Public SkupinaB	Private app ApplicationLocal	
Association Source -> Destination	Public studentTask	Package app ApplicationLocal	
Association Source -> Destination	Public TreeSubjectBean	Package app ApplicationLocal	

Operations

Method	Notes	Parameters
getIPredmetD() IPredmetD Public	Metoda vrati instanci tridy, ktera zprostredkuje pristup do datove vrstvy do tridy IPredmetD @return instance tridy IpredmetD	
getISkupinaD() ISkupinaD Public	Metoda vrati instanci tridy, ktera zprostredkuje pristup do datove vrstvy do tridy ISkupinaD @return instance tridy ISkupinaD	
getIUkolD() IUkolD Public	Metoda vrati instanci tridy, ktera zprostredkuje pristup do datove vrstvy do tridy IUkolD @return instance tridy IUkolD	
getIUzivatelD() IUzivatelD Public	Metoda vrati instanci tridy, ktera zprostredkuje pristup do datove vrstvy do tridy IUzivatelD @return instance tridy UzivatelD	
getIZadaniD() IZadaniD Public	Metoda vrati instanci tridy, ktera zprostredkuje pristup do datove vrstvy do tridy IZadaniD @return instance tridy IZadaniD	

IParalelka

Type: **Interface**
Status: Proposed. Version 1.0. Phase 1.0.
Package: IBussiness   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {8A999726-BA43-45cc-9FE8-753CFF746A39}

Interface tridy IParalelka, specifikuje chovani tridy, ktera bude tento interface implementovat.

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public ParalelkaB	Public IParalelka	

IPredmetB

Type:	**Interface**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	IBussiness *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{5799D67D-73D7-4739-AC19-A4640712DDEC}

Interface IPredmetB specifikuje operace a chovani, ktere bude muset implementovat trida, ktera bude implementovat toto rozhranni. Metody jsou zamereny na praci s predmetem. Trida je zamerena na chovani na bussiness logice

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PredmetB	Private predmetB IPredmetB	
Realization Source -> Destination	Public PredmetB	Public IPredmetB	
Association Source -> Destination	Public GroupBean	Package predmetB IPredmetB	
Association Source -> Destination	Public TreeSubjectBean	Package predmetB IPredmetB	
Association Source -> Destination	Public ParalelkaBean	Package predmetB IPredmetB	

Operations

Method	Notes	Parameters
addParalelka() void Public	Metoda vytvori paralelku na zaklade parametru a preda ji k zapisu do datove vrstvy. @param kodPredmetu - kod predmetu	**String** [in] **kod** - paralelky **Predmet** [in] **subject** **List<Kantor>** [in] **m_kantor** - seznam kantoru, kteri mohou ucit v paralelce **List<Student>** [in] **m_student** - seznam studentu, kteri jsou zapsani do paralelky
addPredmet() void Public	Vytvori predmet podle specifikych vlastnosti, kodu a popisu a preda ho k vlozeni do datove vrstvy.	**String** [in] **kod** **String** [in] **popis** popis
removePredmet() void Public	Metoda preda do datove vrstvy pozadavek k odstraneni predmetu se specifikym kodem.	**String** [in] **kod** kod
showSubjects() List<Predmet> Public	Metodo pozada datovou vrstu o vyhledani predmetu na zaklade jejich kodu. @return List predmetu, ktere maji specificky kod	**String** [in] **kod**
showSubjects() List<Predmet> Public	Metoda pozada datovou vrstvu o seznam vsech predmetu a ty vrati uzivateli v datove strukture List @return List premetu	
showSubjectsByTeacher() List<Predmet> Public	Metoda pozada datovou vrstvu o seznam vsech predmetu, ktere jsou prirazeny konkretnimu uciteli.	**Kantor** [in] **k** k

ISkupinaB

Type: **Interface**
Status: Proposed. Version 1.0. Phase 1.0.
Package: IBussiness *Keywords:*
Detail: Created on 23.11.2011. Last modified on 23.11.2011.
GUID: {F33D77A8-0057-4872-BB6B-838F9EEDEA31}

Interface ISkupinaB specifikuje chovani tridy, ktera bude tento interface impelemntovat. Tento interface se zameruje na operace se skupinami.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public GroupBean	Package skupinaB ISkupinaB	

Realization Source -> Destination	Public SkupinaB	Public ISkupinaB	

Operations

Method	Notes	Parameters
createGroup() void Public	Metoda vytvori skupinu podle zadanych parametru a preda ji k zapisu do datove vrstvy	**Paralelka** [in] **paralelka** **List<Student>** [in] **m_student** **String** [in] **nazevSkupiny** nazevSkupiny

IUkolB

Type:	**Interface**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	IBussiness *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{0E891BB1-1095-4c88-8360-3494F45FBEC4}

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public UkolB	Public IUkolB	
Association Source -> Destination	Public UkolB	Private ukolB IUkolB	

Operations

Method	Notes	Parameters
getUkolyAll() List<Ukol> Public	Metoda pozada datovou vrstu o vsechny ukoly v systemu a vrati je v datove strukture List @return List vsech ukolu v databazi	
getUkolyByTeacher() List<Ukol> Public	Metoda pozada datovou vrstu o vsechny ukoly daneho ucitele a vrati jejich list @return List ukolu danneho ucitele.	**Kantor** [in] **teacher**
odevzdejUkol() void Public	Metoda zpracuje vstupni soubor priradiho k uzivateli a k ukolu ktereho se tyka a takto	**File** [in] **soubor**

	upravenou strukturu preda do datove vrstvy k ulozeni	**String** [in] **loginUzivatele** **Ukol** [in] **ukolOdevzdani** ukolOdevzdani
ohodnotUkol() void Public	Metoda prida bodove a slovni hodnoceni, nebo jen jedno z nich na zǎˇkladǍ› vstupnǍch parametrĹŻ ke konkrǍ©tnǍ©mu Ǎşkolu. Pote pozada datovou vrstvu o vlozeni upravene struktury do databaze.	**Ukol** [in] **ukol** **String** [in] **slovniHodnoceni** **int** [in] **bodyHodnoceni** bodyHodnoceni
uzavriUkol() void Public	Metoda zmeni stav ukolu na stav Ukonceno. Pote pozada datovou vrstvu o zapis upravene struktury do DB	**Ukol** [in] **ukol** ukol **String** [in] **stav**
zadejUkol() void Public	Metoda zada mnozinu ukolu, mnozine studentu, ktere vyucuje urcity kantor. Pote pozada datovou vrstu o vlozeni teto struktury do DB.	**List<Student>** [in] **students** **List<Ukol>** [in] **tasks** tasks **Kantor** [in] **loggedKantor**
zadejUkolJednotlivci() void Public	Metoda priradi studentovy novy ukol a pote pozada datovou vrstu o zapis upravenych dat do DB	**Student** [in] **vypracovatelUkolu** vypracovatelUkolu **String** [in] **nazevZadani** **String** [in] **nazevUkolu**
zadejUkolSkupine() void Public	Metoda priradi ukol skupine studentu a pote zavola datovou vrstu a pozada ji o vlozeni novych udaju	**List<Student>** [in] **vypracovateleUkolu** **String** [in] **nazevZadani** nazevZadani **String** [in] **nazevUkolu**

IUzivatelB

Type:	**Interface**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	IBussiness *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{8C42FC72-A743-4569-B043-D4F05677BBC6}

Interface slouzici pro praci s uzivatelem. Trida se zameruje na praci s uzivatelem na business Ǎşrovni.

Tagged Values
- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public FileBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public UzivatelB	Private uzivatelB IUzivatelB	
Association Source -> Destination	Public CheckStudentBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public LoginBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public UserBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public ParalelkaBean	Package uzivatelB IUzivatelB	
Realization Source -> Destination	Public UzivatelB	Public IUzivatelB	

Operations

Method	Notes	Parameters
addUzivatel() void Public	metoda pro vkladani studentu a kantoru pokud je cokoli nulove tak hazi EJBException pokud je login a email ne-unikatni tak hazi EJBException	**String** [in] **login** **String** [in] **heslo** **String** [in] **jmeno** **String** [in] **prijmeni** **String** [in] **mail** **String** [in] **status** - zda se jedna o studenta ci kantora, pokud "student" tak je to student, cokoli jineho je kantor
addUzivatel() void Public	Metoda vytvori uzivatele, ktere jsou ve vstupnim souboru. OÄŤekÄˇvanÄ˝ formÄˇt je CSV, potÄ© poĹĺĺˇdÄˇ datovou vrstu o zÄˇpis tÄ›chto uĹĺĺˇivatelĹ˝ do DB	**InputStream** [in] **f** f
findUserByLogin() Uzivatel Public	Metoda poĹĺĺˇdÄˇ datovou vrstu o vyhledÄˇnÄ uĹĺĺˇivatele na zÄˇkladÄ› jeho loginu. @return Uzivatele nalezenÄ©ho na zÄˇkladÄ›	**String** [in] **login**

	loginu.	
findUserToLogin() Uzivatel Public		**String** [in] **login**
getAllKantor() List<Kantor> Public	Metoda pozada datovou vrstu o vsechny ucitele v DB @return List vsech ucitele v DB	
getStudenstByKantor() List<Student> Public	Metoda poĹľĂˇdĂˇ datovou vrstu o vyhledĂˇnĂ vĹˇech studentĹŻ, kterĂ© uÄŤĂ kantor. @return List studentĹŻ, vyhledanĂ˝ch podle kantora.	**Kantor** [in] **teacher**
getStudentyAll() List<Student> Public	Metoda pozada datovou vrstu o vsechny studenty v DB. @return List studentu.	
getUzivateleAll() List<Uzivatel> Public	Metoda pozada datovou vrstu o vsechny uzivatele v DB @return List vsech uzivatelu v DB	
modifyUser() void Public	Metoda pozada datovou vrstu o upraveni aktualniho studenta, podle jeho loginu, ktery se nemeni.	**Uzivatel** [in] **user** user
removeUser() void Public	Metoda poĹľĂˇdĂˇ datovou vrstvu o odstranĂ›nĂ z databĂˇze uĹľivatele podle jeho loginu.	**String** [in] **login** login
resetPassword() void Public	Metoda nastavi uzivateli vychozi heslo.	**Uzivatel** [in] **user**
updateUdaje() void Public	Metoda upravi uzivateli na zaklade jeho loginu, jeho e-mail a heslo. Pote pozada datovou vrstu o vlozeni takto upravenych udajou do DB	**String** [in] **login** **String** [in] **mail** **String** [in] **heslo**

IZadaniB

Type: **Interface**
Status: Proposed. Version 1.0. Phase 1.0.
Package: IBussiness   *Keywords:*
Detail: Created on 23.11.2011. Last modified on 23.11.2011.
GUID: {4F28AE40-4F36-465f-ADF4-C054A334861E}

Interface IZadaniB specifikuje chovanĂ a operace tĹ™Ădy, kterĂˇ ho bude implementovat. Interface se zamÄ›Ĺ™uje na prĂˇci a operaci se zadĂˇnĂmi.

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ZadaniB	Private zadaniB IZadaniB	
Realization Source -> Destination	Public ZadaniB	Public IZadaniB	

Operations

Method	Notes	Parameters
getAllKantorTask() List<Zadani> Public	Metoda poĹľĂˇdĂˇ datovou vrstvu o vrĂˇcenĂ vĹˇech zadĂˇnĂ, kterĂˇ jsou vytvoĹ™ena specifickĂ˝m kantorem. @return list zadani podle kantora	**String** [in] **kantorLogin**
modifTaskByKantor() void Public	Metoda pozada datovou vrstu o upraveni mnoziny ukolu, kterou vytvoĹ™il prĂˇvĂ› pĹ™ihlĂˇĹˇenĂ˝ kantor.	**String** [in] **loggedKantorLogin** **List<Zadani>** [in] **tasks**
modifyTask() void Public	Metoda upravĂ zadĂˇnĂ a poĹľĂˇdĂˇ datovou vrstu o vloĹľenĂ tohoto zadĂˇnĂ.	**Date** [in] **deadline** **String** [in] **nazev** **String** [in] **popis** **File** [in] **zadaniFile** **String** [in] **zadavatel** **List<Ukol>** [in] **ukoly** **List<Paralelka>** [in] **paralelka**
vlozZadani() void Public	Metoda vytvoĹ™Ă zadĂˇnĂ nastavĂ mu vĹˇechna potĹ™ebnĂ˝ data a potĂ© poĹľĂˇdĂˇ datovou vrstu o vloĹľenĂ tohoto zadĂˇnĂ.	**Date** [in] **deadline** **String** [in] **nazev** **String** [in] **popis** **File** [in] **zadaniFile** zadaniFile **String** [in] **zadavatel**

Support

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: BL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {1681EE40-779A-46e6-9B7E-7E268A0CDE6E}

Support - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
GUID: {E4573298-6C3A-449d-ADAC-8458AB237ECA}
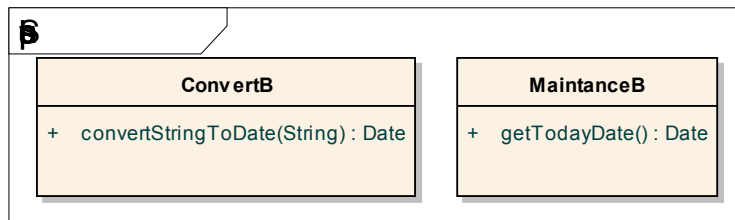
Figure: 4

ConvertB

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Support   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {3F718651-9CFE-4a67-A288-237E9FB46A92}

Trida podporuje chovani aplikace a slouzi jako support pro jednotlive ulohy

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association	Public	Private converter	
Source -> Destination	ZadaniB	ConvertB	

Operations

Method	Notes	Parameters
convertStringToDate() Date Public	Metoda prevede zadany string na datum. @return Date	**String** [in] **datumS**

MaintanceB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Support *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{60DB0782-8FE3-4bff-A633-0258B129CA40}

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
getTodayDate() Date Public	Metoda vrÄˇtÄ aktuÄˇlnÄ datum. @return aktuÄˇlnÄ datum.	

Zadani

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	BL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{4AF9802B-7939-452f-AEEB-CC1BBC7DE862}

Zadani - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{963E59B8-E277-4aca-A2E6-2CCBA6A87168}
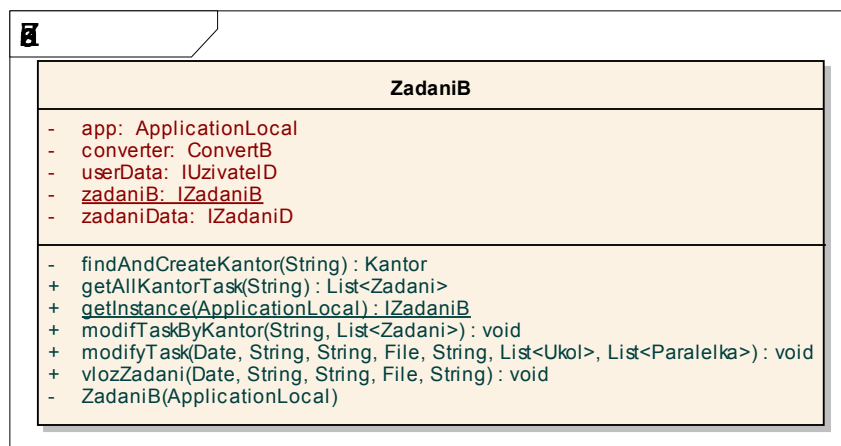
Figure: 5

ZadaniB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Zadani *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{87BF03B9-8F92-451d-BD48-27DD91577B1A}

Singleton trida, ktera implementuje chovani IZadaniB a zprostredkovava sluzby business logiky.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ZadaniB	Private userData IUzivatelD	
Association Source -> Destination	Public ZadaniB	Private zadaniB IZadaniB	
Association Source -> Destination	Public ZadaniB	Private zadaniData IZadaniD	
Association Source -> Destination	Public ZadaniB	Private app ApplicationLocal	
Association Source -> Destination	Public taskBean	Package zadaniBus ZadaniB	
Association Source -> Destination	Public ZadaniB	Private converter ConvertB	
Realization	Public	Public	

Source -> Destination	ZadaniB	IZadaniB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
converter ConvertB Private		*Default:*
userData IUzivatelD Private		*Default:*
zadaniB IZadaniB Private Static		*Default:*
zadaniData IZadaniD Private		*Default:*

Operations

Method	Notes	Parameters
findAndCreateKantor() Kantor Private		**String** [in] **login**
getAllKantorTask()	Metoda poĹľĂˇdĂˇ datovou vrstvu o	**String** [in] **kantorLogin**

List<Zadani> Public	vrĂˇcenĂ vĹˇech zadĂˇnĂ, kterĂˇ jsou vytvoĹ™ena specifickĂ˝m kantorem. @return list zadani podle kantora	
Static **getInstance()** IZadaniB Public		**ApplicationLocal** [in] **app**
modifTaskByKantor() void Public	Metoda pozada datovou vrstu o upraveni mnoziny ukolu, kterou vytvoĹ™il prĂˇvĂ› pĹ™ihlĂˇĹˇenĂ˝ kantor.	**String** [in] **loggedKantorLogin** **List<Zadani>** [in] **tasks**
modifyTask() void Public	Metoda upravĂ zadĂˇnĂ a poĹľĂˇdĂ datovou vrstu o vloĹľenĂ tohoto zadĂˇnĂ.	**Date** [in] **deadline** **String** [in] **nazev** **String** [in] **popis** **File** [in] **zadaniFile** **String** [in] **zadavatel** **List<Ukol>** [in] **ukoly** **List<Paralelka>** [in] **paralelka**
vlozZadani() void Public	Metoda vytvoĹ™Ă zadĂˇnĂ nastavĂ mu vĹˇechna potĹ™ebnĂ˝ data a potĂ© poĹľĂˇdĂ datovou vrstu o vloĹľenĂ tohoto zadĂˇnĂ.	**Date** [in] **deadline** **String** [in] **nazev** **String** [in] **popis** **File** [in] **zadaniFile** zadaniFile **String** [in] **zadavatel**
ZadaniB() Private		**ApplicationLocal** [in] **app**

Predmet

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	BL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{437723E4-A77A-4127-AF12-F1D9D1D4CFC0}

Predmet - *(Logical diagram)*
Created By: papa *on* 23.11.2011

Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
GUID: {500FBF40-A81B-438d-ADEA-6A04AC7B420B}
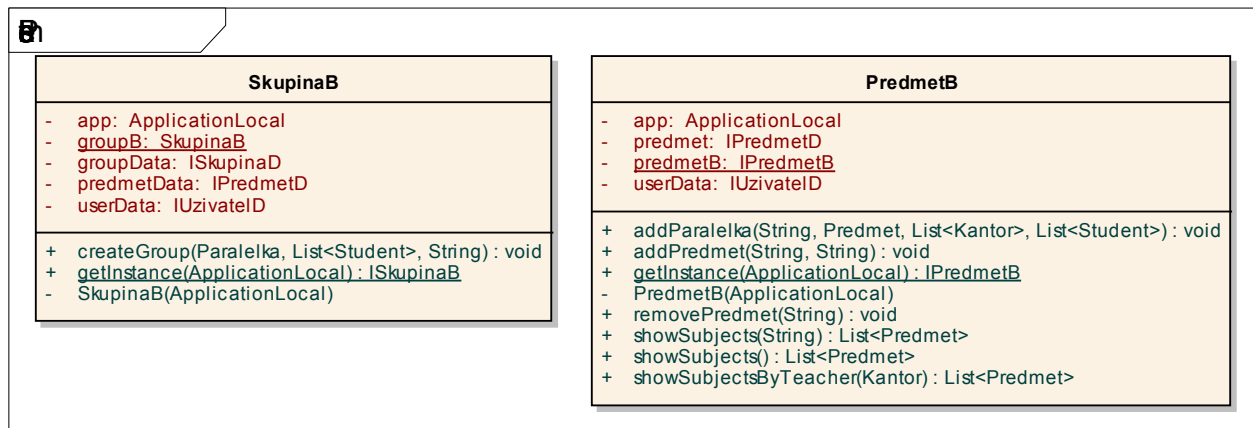

<div style="text-align:center">

SkupinaB

-   app: ApplicationLocal
-   <u>groupB: SkupinaB</u>
-   groupData: ISkupinaD
-   predmetData: IPredmetD
-   userData: IUzivatelD
+   createGroup(Paralelka, List<Student>, String) : void
+   <u>getInstance(ApplicationLocal) : ISkupinaB</u>
-   SkupinaB(ApplicationLocal)

PredmetB

-   app: ApplicationLocal
-   predmet: IPredmetD
-   <u>predmetB: IPredmetB</u>
-   userData: IUzivatelD
+   addParalelka(String, Predmet, List<Kantor>, List<Student>) : void
+   addPredmet(String, String) : void
+   <u>getInstance(ApplicationLocal) : IPredmetB</u>
-   PredmetB(ApplicationLocal)
+   removePredmet(String) : void
+   showSubjects(String) : List<Predmet>
+   showSubjects() : List<Predmet>
+   showSubjectsByTeacher(Kantor) : List<Predmet>

</div>

Figure: 6

PredmetB

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Predmet   *Keywords:*
Detail: Created on 23.11.2011. Last modified on 23.11.2011.
GUID: {5725C2BC-3CF8-4ed6-8572-F2352D8F209B}

Singleton trida, ktera implementuje chovani IPredmetB a zprostredkovava sluzby business logiky. Je primarne zamerana na praci s predmety

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PredmetB	Private app ApplicationLocal	
Association Source -> Destination	Public PredmetB	Private predmet IPredmetD	
Association Source -> Destination	Public PredmetB	Private predmetB IPredmetB	
Association Source -> Destination	Public PredmetB	Private userData IUzivatelD	
Realization	Public	Public	

Source -> Destination	PredmetB	IPredmetB	
Association Source -> Destination	Public AssignTask	Package subjectB PredmetB	
Association Source -> Destination	Public SubjectBean	Package predmetB PredmetB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
predmet IPredmetD Private		*Default:*
predmetB IPredmetB Private Static		*Default:*
userData IUzivatelD Private		*Default:*

Operations

Method	Notes	Parameters
addParalelka() void Public	Metoda vytvori paralelku na zaklade parametru a preda ji k zapisu do datove vrstvy. @param kodPredmetu - kod predmetu	**String** [in] **kod** - paralelky **Predmet** [in] **subject** **List<Kantor>** [in] **m_kantor** - seznam kantoru, kteri mohou ucit

		v paralelce **List\<Student\>** [in] **m_student** - seznam studentu, kteri jsou zapsani do paralelky
addPredmet() void Public	Vytvori predmet podle specifikych vlastnosti, kodu a popisu a preda ho k vlozeni do datove vrstvy.	**String** [in] **kod** **String** [in] **popis** popis
Static **getInstance()** IPredmetB Public		**ApplicationLocal** [in] **app**
PredmetB() Private		**ApplicationLocal** [in] **app**
removePredmet() void Public	Metoda preda do datove vrstvy pozadavek k odstraneni predmetu se specifikym kodem.	**String** [in] **kod** kod
showSubjects() List\<Predmet\> Public	Metodo pozada datovou vrstu o vyhledani predmetu na zaklade jejich kodu. @return List predmetu, ktere maji specificky kod	**String** [in] **kod**
showSubjects() List\<Predmet\> Public	Metoda pozada datovou vrstvu o seznam vsech predmetu a ty vrati uzivateli v datove strukture List @return List premetu	
showSubjectsByTeacher() List\<Predmet\> Public	Metoda pozada datovou vrstvu o seznam vsech predmetu, ktere jsou prirazeny konkretnimu uciteli.	**Kantor** [in] **k** k

SkupinaB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Predmet *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{63790900-DE2A-48a7-A2F1-6A7E4286FD69}

Singleton trida, ktera implementuje chovani ISkupinaB a zprostredkovava sluzby business logiky. Je primarne zamerana na praci se skupinami.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes

Association Source -> Destination	Public SkupinaB	Private app ApplicationLocal	
Association Source -> Destination	Public SkupinaB	Private groupData ISkupinaD	
Association Source -> Destination	Public SkupinaB	Private predmetData IPredmetD	
Association Source -> Destination	Public SkupinaB	Private userData IUzivatelD	
Realization Source -> Destination	Public SkupinaB	Public ISkupinaB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
groupB SkupinaB Private Static		*Default:*
groupData ISkupinaD Private		*Default:*
predmetData IPredmetD Private		*Default:*

userData IUzivatelD Private		Default:

Operations

Method	Notes	Parameters
createGroup() void Public	Metoda vytvori skupinu podle zadanych parametru a preda ji k zapisu do datove vrstvy	**Paralelka** [in] **paralelka** **List<Student>** [in] **m_student** **String** [in] **nazevSkupiny** nazevSkupiny
Static **getInstance()** ISkupinaB Public		**ApplicationLocal** [in] **app**
SkupinaB() Private		**ApplicationLocal** [in] **app**

Paralelka

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: BL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {3E345807-03A9-4ee7-9712-4867A6E1D3EF}

Paralelka - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
GUID: {3C2F9615-01B5-412a-A1FD-F2F7AF636C0C}
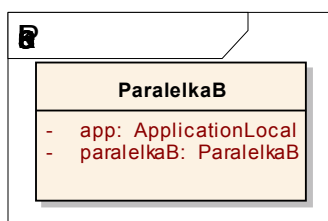
Figure: 7

ParalelkaB

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Paralelka *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{4540D7EF-CF09-47aa-BBA0-76B08964BF95}

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ParalelkaB	Private app ApplicationLocal	
Realization Source -> Destination	Public ParalelkaB	Public IParalelka	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Private		*Default:*
paralelkaB ParalelkaB Private		*Default:*

DL

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Design Model
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{9A8CBBDA-4B34-4863-8525-12849F93813F}

Imanager

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	DL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{3E12DD6C-A1F0-4b19-B36C-B6893DDD0C5E}

Imanager - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{BA4FA053-2BBA-4e39-A8DA-880FFB510645}
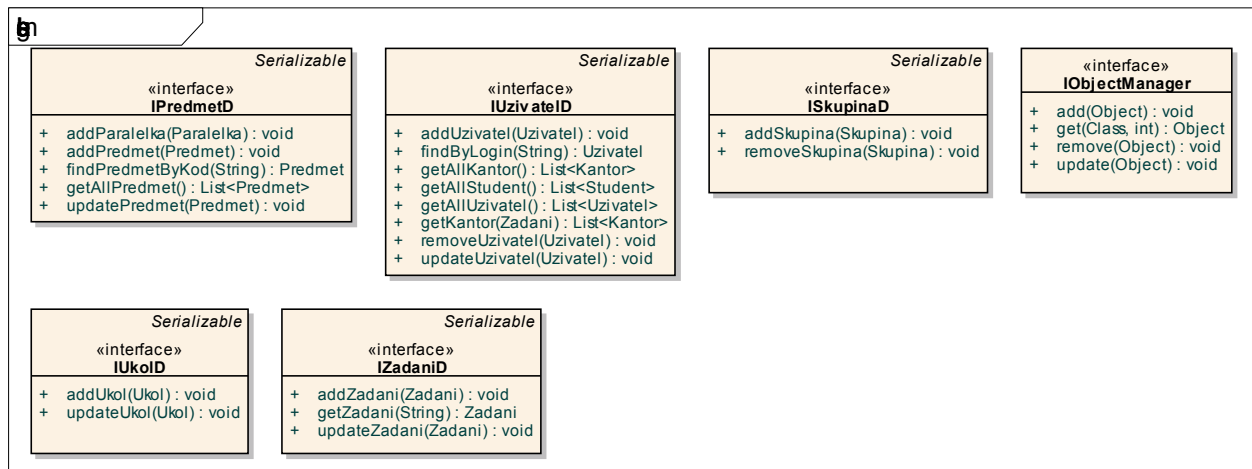
Figure: 8

IObjectManager

Type:	**Interface**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Imanager *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{1E065879-CCEC-4549-8AA7-6C77FD3C1559}

interface pro praci s databazi

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public ObjectManager	Public IObjectManager	

Operations

Method	Notes	Parameters
add() void Public		**Object** [in] **o**
get() Object Public		**Class** [in] **c** **int** [in] **id**
remove() void Public		**Object** [in] **o**
update() void Public		**Object** [in] **o**

IPredmetD

Type:	**Interface Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Imanager *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{063C2183-6988-4201-AB14-C4E296204BE1}

lokalni rozhrani v datove vrstve pro predmety

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public PredmetB	Private predmet IPredmetD	
Realization Source -> Destination	Public PredmetD	Public IPredmetD	
Association Source -> Destination	Public Application	Private predD IPredmetD	
Association Source -> Destination	Public SkupinaB	Private predmetData IPredmetD	

Operations

Method	Notes	Parameters

addParalelka() void Public		**Paralelka** [in] **p**
addPredmet() void Public		**Predmet** [in] **p**
findPredmetByKod() Predmet Public		**String** [in] **kod**
getAllPredmet() List<Predmet> Public		
updatePredmet() void Public		**Predmet** [in] **p**

ISkupinaD

Type:	**Interface Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Imanager *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{5F51DF46-5250-4588-82B9-1924B99BB95F}

lokalni rozhrani v datove vrstve pro skupiny v paralelkach

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public SkupinaD	Public ISkupinaD	
Association Source -> Destination	Public SkupinaB	Private groupData ISkupinaD	
Association Source -> Destination	Public Application	Private skupD ISkupinaD	

Operations

Method	Notes	Parameters
addSkupina() void Public		**Skupina** [in] **s**
removeSkupina() void Public		**Skupina** [in] **s**

IUkolD

Type:	**Interface Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Imanager *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{D595D2E3-0873-4e69-B3D1-D7A21139ACE7}

lokalni rozhrani v datove vrstve pro ukoly

Tagged Values
- annotations = @Local.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public UkolD	Public IUkolD	
Association Source -> Destination	Public UkolB	Private ukolData IUkolD	
Association Source -> Destination	Public testBean	Package ukolD IUkolD	
Association Source -> Destination	Public Application	Private ukolD IUkolD	

Operations

Method	Notes	Parameters
addUkol() void Public	prida ukol do db	**Ukol** [in] **u** - novy ukol
updateUkol() void Public	upravi ukol do db	**Ukol** [in] **ukol** - upravovany ukol

IUzivatelD

Type:	**Interface Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Imanager *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{665BA42D-71DD-475e-A51B-8A825BEAE77E}

lokalni rozhrani v datove vrstve pro uzivatele

Tagged Values
- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ZadaniB	Private userData IUzivatelD	
Association Source -> Destination	Public PredmetB	Private userData IUzivatelD	
Realization Source -> Destination	Public UzivatelD	Public IUzivatelD	
Association Source -> Destination	Public UzivatelB	Private userData IUzivatelD	
Association Source -> Destination	Public testBean	Package uzivD IUzivatelD	
Association Source -> Destination	Public Application	Private uzivD IUzivatelD	
Association Source -> Destination	Public SkupinaB	Private userData IUzivatelD	

Operations

Method	Notes	Parameters
addUzivatel() void Public	ulozi uzivatele do db	**Uzivatel** [in] **u** u
findByLogin() Uzivatel Public		**String** [in] **login**
getAllKantor() List<Kantor> Public		
getAllStudent() List<Student> Public		
getAllUzivatel() List<Uzivatel> Public		
getKantor() List<Kantor> Public	vrati vsechny kantory, kteri vytvorily zadani @return	**Zadani** [in] **zadani**

removeUzivatel() void Public		**Uzivatel** [in] **u**
updateUzivatel() void Public		**Uzivatel** [in] **u**

IZadaniD

Type: **Interface   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Imanager   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {E564CC81-5023-4a7d-84D5-1D6BD2AEBE67}

lokalni rozhrani v datove vrstve pro zadani

Tagged Values

- annotations = @Local.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public ZadaniB	Private zadaniData IZadaniD	
Association Source -> Destination	Public UkolB	Private zadaniData IZadaniD	
Realization Source -> Destination	Public ZadaniD	Public IZadaniD	
Association Source -> Destination	Public testBean	Package zadaD IZadaniD	
Association Source -> Destination	Public Application	Private zadaD IZadaniD	

Operations

Method	Notes	Parameters
addZadani() void Public	ulozi nove zadani do db	**Zadani** [in] **z** z
getZadani() Zadani Public	vrati zadani podle jejiho nazvu @return	**String** [in] **nazev**
updateZadani() void Public	upravi zmenene zadani do db	**Zadani** [in] **zadani** zadani

entity

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	DL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{19D540FE-D1B7-4445-8049-A26616A01A3A}

entity - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{EA25CF30-5A32-4332-B710-EB4817B1AD91}
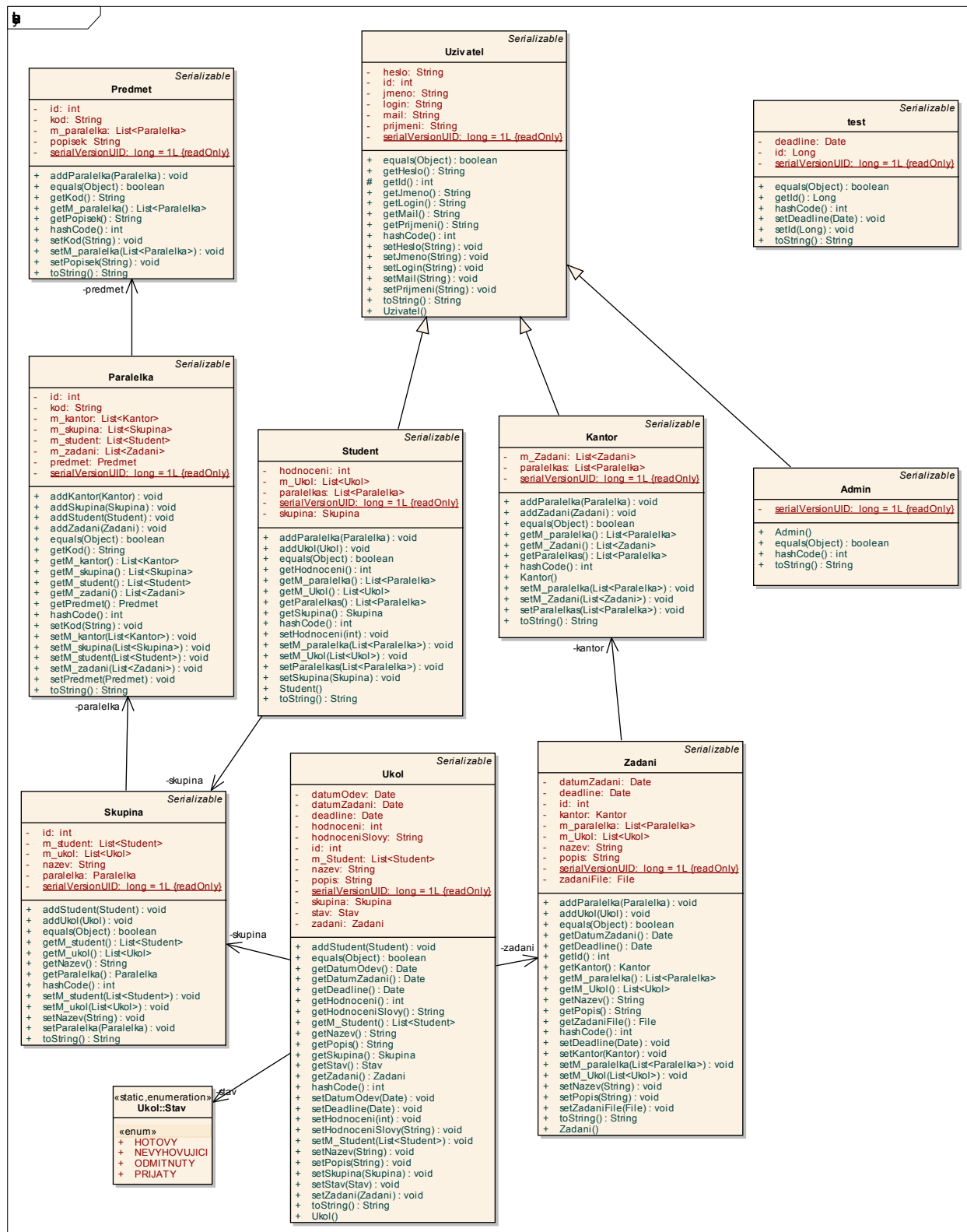
Figure: 9

Admin

Type:	**Class Uzivatel, Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{AE1D8269-7A9E-4a1c-ACD5-EEE8EAF23D0B}

trida reprezentujici admina

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "admin")@DiscriminatorValue(value = "admin")@PrimaryKeyJoinColumn(name = "id").

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Admin	Public Uzivatel	

Attributes

Attribute	Notes	Constraints and tags
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
Admin() Public		
equals() boolean Public		**Object** [in] **object**
hashCode() int Public		
toString() String Public		

Kantor

Type:	**Class Uzivatel, Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{3773CE35-FC31-469a-82D2-1708EC7E2D44}

trida reprezentujici kantora(ucitele)

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "kantor")@DiscriminatorValue(value = "kantor")@PrimaryKeyJoinColumn(name = "id")@NamedQueries({
    @NamedQuery(name = "Kantor.findByZadani", query = "SELECT k FROM Kantor k WHERE k.m_Zadani = :m_Zadani"),
    @NamedQuery(name = "Kantor.all", query = "SELECT k FROM Kantor k")
  }).

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AssignTask	Package loggedKantor Kantor	
Generalization Source -> Destination	Public Kantor	Public Uzivatel	
Association Source -> Destination	Public Zadani	Private kantor Kantor	
Association Source -> Destination	Public ParalelkaBean	Package vybraniKantori Kantor	
Association Source -> Destination	Public studentTask	Package loggedKantor Kantor	

Attributes

Attribute	Notes	Constraints and tags

m_Zadani List<Zadani> Private		*Default:* [annotations = @OneToMany(mappedBy = "kantor", fetch = FetchType.EAGER, cascade = CascadeType.ALL)]
paralelkas List<Paralelka> Private		*Default:* [annotations = @ManyToMany(mappedBy = "m_kantor")]
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
addParalelka() void Public	prida paralelku kantorovi	**Paralelka** [in] **p** p
addZadani() void Public	prida dalsi zadani od kantora	**Zadani** [in] **z** z
equals() boolean Public		**Object** [in] **object**
getM_paralelka() List<Paralelka> Public		
getM_Zadani() List<Zadani> Public		
getParalelkas() List<Paralelka> Public		
hashCode() int Public		
Kantor() Public		
setM_paralelka() void Public		**List<Paralelka>** [in] **m_paralelka**
setM_Zadani() void Public		**List<Zadani>** [in] **m_Zadani**

setParalelkas() void Public		**List<Paralelka>** [in] **paralelkas**
toString() String Public		

Paralelka

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{2F9C189E-887E-4264-9422-C7B561D6493C}

trida reprezentujici paralelku

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "paralelka",
  uniqueConstraints=@UniqueConstraint(columnNames={"predmet_id", "kod"})).

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public GroupBean	Package paralelkaChosen Paralelka	
Association Source -> Destination	Public Paralelka	Private predmet Predmet	
Association Source -> Destination	Public Skupina	Private paralelka Paralelka	
Association Source -> Destination	Public AssignTask	Package paralelChosen Paralelka	
Association Source -> Destination	Public AssignTask	Package paralelkaAsSubject Paralelka	
Association Source -> Destination	Public AssignTask	Package temp Paralelka	

Attributes

Attribute	Notes	Constraints and tags
id int Private		*Default:* [annotations = @Id@GeneratedValue(strategy = GenerationType.AUTO)]
kod String Private		*Default:* [annotations = @Column(name = "kod", nullable = false)]
m_kantor List<Kantor> Private		*Default:* [annotations = @ManyToMany]
m_skupina List<Skupina> Private		*Default:* [annotations = @OneToMany(mappedBy = "paralelka")]
m_student List<Student> Private		*Default:* [annotations = @ManyToMany]
m_zadani List<Zadani> Private		*Default:* [annotations = @ManyToMany]

predmet Predmet Private		*Default:* [annotations = @JoinColumn(name="predmet_id") @ManyToOne(cascade = CascadeType.ALL, fetch = FetchType.EAGER)]
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
addKantor() void Public	prida kantora do paralelky	**Kantor** [in] **k** - ten kdo uci tuto paralelku
addSkupina() void Public	prida novou skupinu do paralelky	**Skupina** [in] **s** - nova skupina
addStudent() void Public	prida studenta do paralelky	**Student** [in] **s** - novy student
addZadani() void Public	prida zadani do paralelky	**Zadani** [in] **z** - nove zadani
equals() boolean Public		**Object** [in] **object**
getKod() String Public		
getM_kantor() List<Kantor> Public		
getM_skupina() List<Skupina> Public		
getM_student() List<Student> Public		
getM_zadani() List<Zadani> Public		
getPredmet() Predmet Public		
hashCode() int Public		
setKod() void Public		**String** [in] **kod**

setM_kantor() void Public		**List<Kantor>** [in] **m_kantor**
setM_skupina() void Public		**List<Skupina>** [in] **m_skupina**
setM_student() void Public		**List<Student>** [in] **m_student**
setM_zadani() void Public		**List<Zadani>** [in] **m_zadani**
setPredmet() void Public		**Predmet** [in] **predmet**
toString() String Public		

Predmet

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: entity   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {AD85249D-1B7B-4623-ADC5-1EF36EDCF28E}

trida reprezentujici predmet

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "predmet")@NamedQueries({
    @NamedQuery(name = "Predmet.findByKod", query = "SELECT p FROM Predmet p WHERE p.kod = :kod"),
    @NamedQuery(name = "Predmet.all", query = "SELECT p FROM Predmet p")
  }).

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AssignTask	Package kantrosSubjectToFilter Predmet	
Association Source -> Destination	Public GroupBean	Package predmetChosen	

		Predmet	
Association Source -> Destination	Public Paralelka	Private predmet Predmet	
Association Source -> Destination	Public ParalelkaBean	Package vybranyPredmet Predmet	

Attributes

Attribute	Notes	Constraints and tags
id int Private		*Default:* [annotations = @Id@GeneratedValue(strategy = GenerationType.AUTO)]
kod String Private		*Default:* [annotations = @Column(name = "kod", nullable = false, unique = true)]
m_paralelka List<Paralelka> Private		*Default:* [annotations = @OneToMany(mappedBy = "predmet", cascade = CascadeType.ALL, fetch = FetchType.EAGER)]
popisek String Private		*Default:* [annotations = @Column(name = "popisek")]
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
addParalelka() void Public	prida novou paralelku	**Paralelka** [in] **p** - nova paralelka
equals() boolean Public		**Object** [in] **object**
getKod() String Public		
getM_paralelka() List<Paralelka> Public		
getPopisek() String Public		
hashCode() int Public		
setKod() void Public		**String** [in] **kod**
setM_paralelka() void Public		**List<Paralelka>** [in] **m_paralelka**
setPopisek() void Public		**String** [in] **popisek**
toString() String Public		

Skupina

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{987368BB-9F68-420e-8C8C-4D0202A71666}

trida reprezentujici skupiny v paralelce

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "skupina").

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Student	Private skupina Skupina	
Association Source -> Destination	Public Ukol	Private skupina Skupina	
Association Source -> Destination	Public Skupina	Private paralelka Paralelka	

Attributes

Attribute	Notes	Constraints and tags
id int Private		*Default:* [annotations = @Id@GeneratedValue(strategy = GenerationType.AUTO)]
m_student List<Student> Private		*Default:* [annotations = @OneToMany(mappedBy = "skupina")]
m_ukol List<Ukol> Private		*Default:* [annotations = @OneToMany(mappedBy = "skupina")]
nazev String Private		*Default:* [annotations = @Column(name = "nazev", nullable = false, unique = true)]
paralelka Paralelka Private		*Default:* [annotations = @ManyToOne]

serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
addStudent() void Public	prida studenta do skupiny	**Student** [in] **s** - novy student
addUkol() void Public	prida novy ukol skupine	**Ukol** [in] **u** - novy ukol
equals() boolean Public		**Object** [in] **object**
getM_student() List<Student> Public		
getM_ukol() List<Ukol> Public		
getNazev() String Public		
getParalelka() Paralelka Public		
hashCode() int Public		
setM_student() void Public		**List<Student>** [in] **m_student**
setM_ukol() void Public		**List<Ukol>** [in] **m_ukol**
setNazev() void Public		**String** [in] **nazev**
setParalelka() void Public		**Paralelka** [in] **paralelka**
toString() String Public		

Student

Type:	**Class Uzivatel, Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{09250F25-B382-4b80-94A6-1E434E05C156}

trida reprezentujici studenta

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "student")@DiscriminatorValue(value = "student")@PrimaryKeyJoinColumn(name = "id")@NamedQueries({
    //   @NamedQuery(name = "Uzivatel.findByLogin", query = "SELECT u FROM Uzivatel u WHERE u.login = :login"),
    @NamedQuery(name = "Student.all", query = "SELECT s FROM Student s")
  }).

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AssignTask	Package currChosen Student	
Association Source -> Destination	Public Student	Private skupina Skupina	
Generalization Source -> Destination	Public Student	Public Uzivatel	
Association Source -> Destination	Public studentTask	Package student Student	
Association Source -> Destination	Public ParalelkaBean	Package vybraniStudenti Student	

Attributes

Attribute	Notes	Constraints and tags
hodnoceni int Private		*Default:* [annotations = @Column(name = "hodnoceni")]

m_Ukol List<Ukol> Private		*Default:* [annotations = @ManyToMany(mappedBy = "m_Student", fetch= FetchType.EAGER, cascade= CascadeType.ALL)]
paralelkas List<Paralelka> Private		*Default:* [annotations = @ManyToMany(mappedBy = "m_student")]
serialVersionUID long Private Static Const		*Default:* 1L
skupina Skupina Private		*Default:* [annotations = @ManyToOne]

Operations

Method	Notes	Parameters
addParalelka() void Public	prida studentovi novou paralelku	**Paralelka** [in] **p** - nova paralelka
addUkol() void Public	prida studentovi novy ukol	**Ukol** [in] **u** - novy ukol
equals() boolean Public		**Object** [in] **object**
getHodnoceni() int Public		
getM_paralelka() List<Paralelka> Public		
getM_Ukol() List<Ukol> Public		
getParalelkas() List<Paralelka> Public		

getSkupina() Skupina Public		
hashCode() int Public		
setHodnoceni() void Public		**int** [in] **newVal** newVal
setM_paralelka() void Public		**List<Paralelka>** [in] **m_paralelka**
setM_Ukol() void Public		**List<Ukol>** [in] **m_Ukol**
setParalelkas() void Public		**List<Paralelka>** [in] **paralelkas**
setSkupina() void Public		**Skupina** [in] **skupina**
Student() Public		
toString() String Public		

Ukol

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: entity   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {8A9C7782-1D50-47de-92B9-43B022BA10EF}

ukoly, ktere studenti/skupiny studentu vypracovavaji

Custom Properties
- isActive = False

Tagged Values
- annotations = @Entity@Table(name = "ukol").

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Ukol	Private skupina Skupina	
Association Source -> Destination	Public Ukol	Private stav Stav	

Association Source -> Destination	Public Ukol	Private zadani Zadani	

Attributes

Attribute	Notes	Constraints and tags
datumOdev Date Private		*Default:* [annotations = @Column(name = "datumOdev")@Temporal(Tempora lType.DATE)]
datumZadani Date Private		*Default:* [annotations = @Column(name = "datumZadani", nullable = false)@Temporal(TemporalType.DA TE)]
deadline Date Private		*Default:* [annotations = @Column(name = "deadline")@Temporal(TemporalTy pe.DATE)]
hodnoceni int Private		*Default:* [annotations = @Column(name = "hodnoceni")]
hodnoceniSlovy String Private		*Default:* [annotations = @Column(name = "hodnoceniSlovy")]

id int Private		*Default:* [annotations = @GeneratedValue@Id@Column(name = "id")]
m_Student List<Student> Private		*Default:* [annotations = @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)]
nazev String Private		*Default:* [annotations = @Column(name = "nazev")]
popis String Private		*Default:* [annotations = @Column(name = "popis")]
serialVersionUID long Private Static Const		*Default:* 1L
skupina Skupina Private		*Default:* [annotations = @ManyToOne]

stav Stav Private		*Default:* [annotations = @Column(name = "stav")]
zadani Zadani Private		*Default:* [annotations = @ManyToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)]

Operations

Method	Notes	Parameters
addStudent() void Public	ukolu priradi studenta	**Student** [in] **s** - student
equals() boolean Public		**Object** [in] **object**
getDatumOdev() Date Public		
getDatumZadani() Date Public		
getDeadline() Date Public		
getHodnoceni() int Public		
getHodnoceniSlovy() String Public		
getM_Student() List<Student> Public		
getNazev() String Public		
getPopis() String Public		
getSkupina() Skupina Public		
getStav() Stav Public		
getZadani() Zadani Public		
hashCode() int Public		
setDatumOdev() void Public		**Date** [in] **newVal**

setDeadline() void Public		**Date** [in] **deadline**
setHodnoceni() void Public		**int** [in] **newVal**
setHodnoceniSlovy() void Public		**String** [in] **newVal**
setM_Student() void Public		**List<Student>** [in] **m_Student**
setNazev() void Public		**String** [in] **newVal**
setPopis() void Public		**String** [in] **popis**
setSkupina() void Public		**Skupina** [in] **skupina**
setStav() void Public		**Stav** [in] **newVal**
setZadani() void Public		**Zadani** [in] **zadani**
toString() String Public		
Ukol() Public		

Stav

Type:	**Enumeration**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{0A7BF0E9-E147-4953-99F3-6F592B424C49}

v jakem stavu se nachazi ukol

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Ukol	Private stav Stav	

Attributes

Attribute	Notes	Constraints and tags
HOTOVY Public «enum»	hotovy - student vypracoval ukol a ten ceka na revizi	*Default:*
NEVYHOVUJICI Public «enum»	nevyhovujici - kantro ho zkontroloval, ale studenti jej musi predelat	*Default:*
ODMITNUTY Public «enum»	odmitnuty - odevzdano pozde	*Default:*
PRIJATY Public «enum»	prijaty - kantor pridelil ukol studentovi	*Default:*

Uzivatel

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{FFC3A4D6-7EB2-4651-BDD6-C3CFC795B5DC}

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "uzivatel")@Inheritance(strategy = InheritanceType.JOINED)@DiscriminatorColumn(name = "role", discriminatorType = DiscriminatorType.STRING, length = 8)@NamedQueries({
    @NamedQuery(name = "Uzivatel.findByLogin", query = "SELECT u FROM Uzivatel u WHERE u.login = :login"),
    @NamedQuery(name = "Uzivatel.all", query = "SELECT u FROM Uzivatel u")//,
  //   @NamedQuery(name="Uzivatel.findByZadani", query="SELECT u FROM Uzavatel u")
  }).

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public Admin	Public Uzivatel	
Association Source -> Destination	Public LoginBean	Package loggedPerson Uzivatel	
Generalization Source -> Destination	Public Kantor	Public Uzivatel	
Generalization Source -> Destination	Public Student	Public Uzivatel	
Association Source -> Destination	Public CheckStudentBean	Package chosen Uzivatel	

Attributes

Attribute	Notes	Constraints and tags
heslo String Private		*Default:* [annotations = @Column(name = "heslo", nullable = false)]
id int Private		*Default:* [annotations = @Id@GeneratedValue@Column(name = "id")]

jmeno String Private		*Default:* [<u>annotations</u> = @Column(name = "jmeno", nullable = false)]
login String Private		*Default:* [<u>annotations</u> = @Column(name = "login", nullable = false, unique = true)]
mail String Private		*Default:* [<u>annotations</u> = @Column(name = "mail", nullable = false, unique = true)]
prijmeni String Private		*Default:* [<u>annotations</u> = @Column(name = "prijmeni", nullable = false)]
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
equals() boolean Public		**<u>Object</u>** [in] **<u>object</u>**
getHeslo() String Public		
getId() int Protected		
getJmeno() String Public		
getLogin() String		

Public		
getMail() String Public		
getPrijmeni() String Public		
hashCode() int Public		
setHeslo() void Public		**String** [in] **newVal**
setJmeno() void Public		**String** [in] **newVal**
setLogin() void Public		**String** [in] **newVal**
setMail() void Public		**String** [in] **newVal**
setPrijmeni() void Public		**String** [in] **newVal**
toString() String Public		
Uzivatel() Public		

Zadani

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: entity   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {72CEF06B-8411-4a78-BE00-33D50388E3E0}

zadani, ktere kantor vytvoril studentum/skupinam studentu, studenti/skupiny pak maji vlastni ukoly, na kterych
pracuji

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity@Table(name = "zadani")@NamedQueries({
    @NamedQuery(name = "Zadani.findByNazev", query = "SELECT z FROM Zadani z WHERE z.nazev
  = :nazev")
  }).

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public Zadani	Private kantor Kantor	
Association Source -> Destination	Public Ukol	Private zadani Zadani	
Association Source -> Destination	Public AssignTask	Package taskChosen Zadani	

Attributes

Attribute	Notes	Constraints and tags
datumZadani Date Private		*Default:* [annotations = @Column(name = "datumZadani")@Temporal(TemporalType.DATE)]
deadline Date Private		*Default:* [annotations = @Column(name = "deadline")@Temporal(TemporalType.DATE)]
id int Private		*Default:* [annotations = @Id@GeneratedValue@Column(name = "id")]
kantor Kantor Private		*Default:* [annotations = @ManyToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)]

m_paralelka List<Paralelka> Private		*Default:* [<u>annotations</u> = @ManyToMany(mappedBy = "m_zadani")]
m_Ukol List<Ukol> Private		*Default:* [<u>annotations</u> = @OneToMany(mappedBy = "zadani", fetch = FetchType.EAGER, cascade = CascadeType.ALL)]
nazev String Private		*Default:* [<u>annotations</u> = @Column(name = "nazev")]
popis String Private		*Default:* [<u>annotations</u> = @Column(name = "popis")]
serialVersionUID long Private Static Const		*Default:* 1L
zadaniFile File Private		*Default:* [<u>annotations</u> = @Column(name = "zadaniFile")]

Operations

Method	Notes	Parameters
addParalelka() void Public	prida paralelku k zadani	**Paralelka** [in] **p** - paralelka

addUkol() void Public	prida novy ukol k zadani	**Ukol** [in] **u** - novy ukol
equals() boolean Public		**Object** [in] **object**
getDatumZadani() Date Public		
getDeadline() Date Public		
getId() int Public		
getKantor() Kantor Public		
getM_paralelka() List<Paralelka> Public		
getM_Ukol() List<Ukol> Public		
getNazev() String Public		
getPopis() String Public		
getZadaniFile() File Public		
hashCode() int Public		
setDeadline() void Public		**Date** [in] **deadline**
setKantor() void Public		**Kantor** [in] **kantor**
setM_paralelka() void Public		**List<Paralelka>** [in] **m_paralelka**
setM_Ukol() void Public		**List<Ukol>** [in] **m_Ukol**
setNazev() void Public		**String** [in] **newVal**
setPopis() void Public		**String** [in] **newVal**
setZadaniFile() void Public		**File** [in] **newVal**
toString() String Public		
Zadani() Public		

test

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	entity *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{2191A1E9-24DD-4de0-B024-96BB17EA5458}

Custom Properties

- isActive = False

Tagged Values

- annotations = @Entity.

Attributes

Attribute	Notes	Constraints and tags
deadline Date Private		*Default:* [annotations = @Temporal(TemporalType.DATE)]
id Long Private		*Default:* [annotations = @Id@GeneratedValue(strategy = GenerationType.AUTO)]
serialVersionUID long Private Static Const		*Default:* 1L

Operations

Method	Notes	Parameters
equals() boolean Public		**Object** [in] **object**

getId() Long Public		
hashCode() int Public		
setDeadline() void Public		**Date** [in] **deadline**
setId() void Public		**Long** [in] **id**
toString() String Public		

manager

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	DL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{0AE81C2C-7A4F-4bc8-8085-7E08294340AD}

manager - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{45AF3682-0CF4-4302-911B-5B3FE9A211B4}

Figure: 10
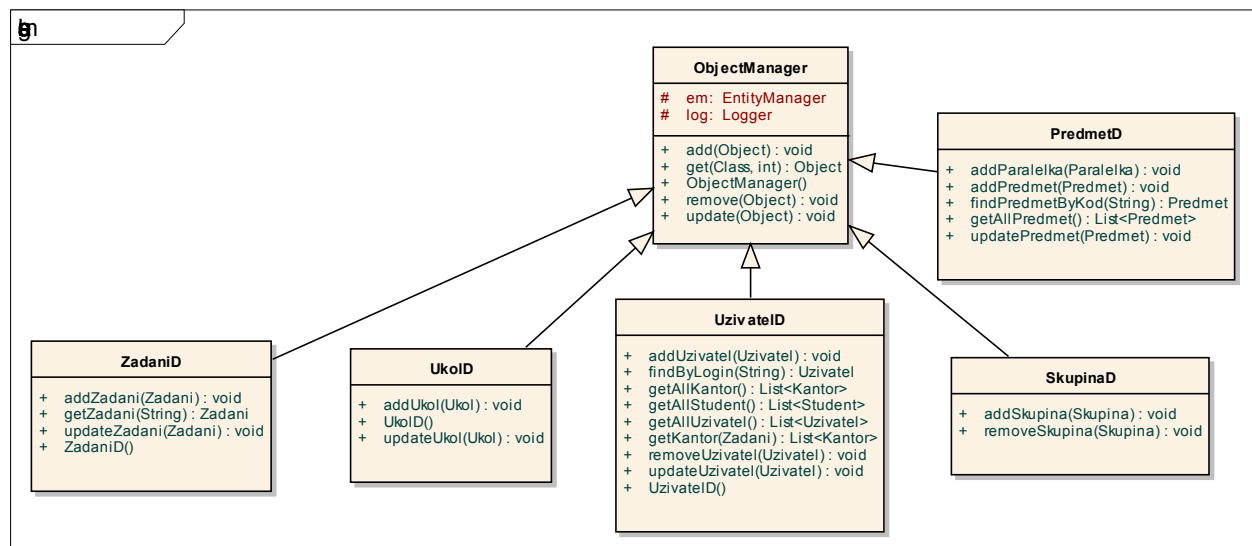
ObjectManager

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.

Package: manager   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {ABE63445-0055-4314-8FF8-45725C20FFF2}

implementace IObjectManager pro praci s databazi na nejuniverzalnejsi urovni

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public ZadaniD	Public ObjectManager	
Generalization Source -> Destination	Public UkolD	Public ObjectManager	
Generalization Source -> Destination	Public UzivatelD	Public ObjectManager	
Generalization Source -> Destination	Public SkupinaD	Public ObjectManager	
Generalization Source -> Destination	Public PredmetD	Public ObjectManager	
Realization Source -> Destination	Public ObjectManager	Public IObjectManager	

Attributes

Attribute	Notes	Constraints and tags
em EntityManager Protected		*Default:* [annotations = @PersistenceContext(unitName = "SZU-ejbPU")]
log Logger Protected		*Default:*

Operations

Method	Notes	Parameters
add() void Public	ulozi objekt do db	**Object** [in] **o** o
get() Object Public	vrati objekt z db @return	**Class** [in] **c** **int** [in] **id**
ObjectManager() Public		
remove() void Public	odstrani objekt z db	**Object** [in] **o** o
update() void Public	vlozi upraveny objekt z db	**Object** [in] **o** o

PredmetD

Type:	**Class ObjectManager**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	manager *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{34F94F26-6D78-4369-A192-C6EF3C7C8B39}

manager pro praci s predmety v datove vrstve

Custom Properties
- isActive = False

Tagged Values
- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public PredmetD	Public ObjectManager	
Realization Source -> Destination	Public PredmetD	Public IPredmetD	

Operations

Method	Notes	Parameters
addParalelka() void Public	prida paralelku do predmetu	**Paralelka** [in] **p** - nova paralelka
addPredmet() void Public	prida predmet	**Predmet** [in] **p** - novy predmet
findPredmetByKod() Predmet Public	najde predmet dle jeho kodu @return predmet se zadanym kodem	**String** [in] **kod** - kod predmetu
getAllPredmet() List<Predmet> Public	vrati vsechny predmety @return - vsechny predmety	
updatePredmet() void Public	upraveny predmet ulozi do db	**Predmet** [in] **p** - upraveny predmet

SkupinaD

Type: **Class   ObjectManager**
Status: Proposed. Version 1.0. Phase 1.0.
Package: manager   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {9602E1C1-CD3F-43cb-9078-FE41390B9B5F}

manager pro praci s skupinami v datove vrstve

Custom Properties

- isActive = False

Tagged Values

- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public SkupinaD	Public ObjectManager	
Realization Source -> Destination	Public SkupinaD	Public ISkupinaD	

Operations

Method	Notes	Parameters
addSkupina() void Public	prida novou skupinu	**Skupina** [in] **s** - nova skupina

removeSkupina() void Public	odstrani skupinu	**Skupina** [in] **s** - skupina k odstraneni

UkolD

Type: **Class   ObjectManager**
Status: Proposed. Version 1.0. Phase 1.0.
Package: manager   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {358B5BF7-CFF5-4904-A32E-BF6EFBEE72C0}

manager pro praci s ukoly v datove vrstve

Custom Properties

- isActive = False

Tagged Values

- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public UkolD	Public IUkolD	
Generalization Source -> Destination	Public UkolD	Public ObjectManager	

Operations

Method	Notes	Parameters
addUkol() void Public	prida ukol do db	**Ukol** [in] **u** - novy ukol
UkolD() Public		
updateUkol() void Public	upraveny ukol ulozi do db	**Ukol** [in] **ukol** - upraveny ukol

UzivatelD

Type: **Class   ObjectManager**
Status: Proposed. Version 1.0. Phase 1.0.

Package: manager   *Keywords:*
Detail: Created on 23.11.2011. Last modified on 23.11.2011.
GUID: {B743F8DC-F770-44af-A071-3BACC23714EE}

manager pro praci s uzivateli v datove vrstve

Custom Properties

- isActive = False

Tagged Values

- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public UzivatelD	Public ObjectManager	
Realization Source -> Destination	Public UzivatelD	Public IUzivatelD	

Operations

Method	Notes	Parameters
addUzivatel() void Public	prida uzivatele do db	**Uzivatel** [in] **u** u
findByLogin() Uzivatel Public	vyhleda uzivatele podle loginu @return vyhledany uzivatel	**String** [in] **login** - login uzivatele
getAllKantor() List<Kantor> Public	vrati vsechny kantory @return	
getAllStudent() List<Student> Public	vrati vsechny studenty @return	
getAllUzivatel() List<Uzivatel> Public	vrati vsechny uzivatele @return	
getKantor() List<Kantor> Public	vrati kantory, kteri vytvorily zadani @return list kantoru	**Zadani** [in] **zadani** - zadani asociovane s kantorem
removeUzivatel() void Public	odstrani uzivatele z db	**Uzivatel** [in] **u** u
updateUzivatel() void Public	upravi uzivatele	**Uzivatel** [in] **u** - upraveny uzivatel
UzivatelD()		

Public		

ZadaniD

Type:	**Class ObjectManager**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	manager *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{D4CF7924-D080-498c-8582-DD9D87D346E3}

manager pro praci s zadanimi v datove vrstve

Custom Properties

- isActive = False

Tagged Values

- annotations = @Stateless.

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public ZadaniD	Public ObjectManager	
Realization Source -> Destination	Public ZadaniD	Public IZadaniD	

Operations

Method	Notes	Parameters
addZadani() void Public	prida zadani do db	**Zadani** [in] **z** - nove zadani
getZadani() Zadani Public	vrati zadani s danym nazvem @return zadani s danym nazvem	**String** [in] **nazev** - nazev zadani
updateZadani() void Public	upravi zadani do db	**Zadani** [in] **zadani** - upravene zadani
ZadaniD() Public		

PL

Type:	**Package**

Package: Design Model
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {59097622-C27C-459b-A530-877982BD1DB1}

backingBean

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: PL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {211A9D10-4B5A-450b-8FCD-2F260D2E1A4E}

backingBean - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
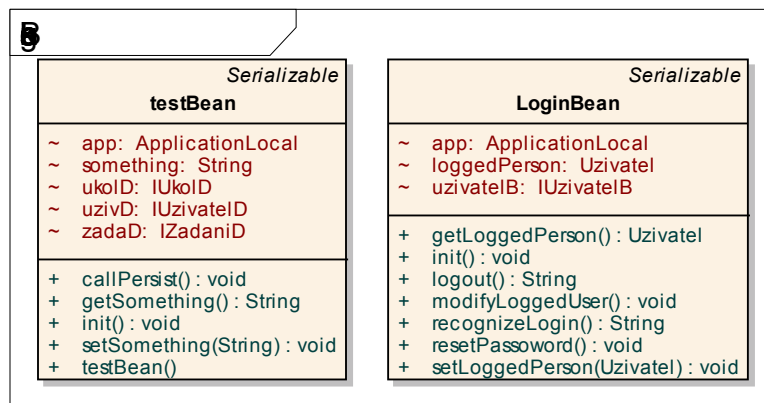*GUID:* {41193B19-2C67-4eb7-A1A7-8FB3450C4AE1}

Figure: 11

LoginBean

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: backingBean   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {415AB68A-40B7-46ca-8C88-F611DF18E396}

Beana, ktera si udrzuje informace o aktualne prihlasem uzivateli. Zprostredkovava logout z frameworku JAAS

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name="loginBean")@SessionScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public LoginBean	Package loggedPerson Uzivatel	
Association Source -> Destination	Public LoginBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public LoginBean	Package app ApplicationLocal	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
loggedPerson Uzivatel Package		*Default:*
uzivatelB IUzivatelB Package	instance v bussiness vrstve starajici se o uzivatele	*Default:*

Operations

Method	Notes	Parameters
getLoggedPerson() Uzivatel Public		
init() void Public		
logout() String	Metoda odhlasi uzivatele z aplikace a udela	

Public	ivalidate sessions.	
modifyLoggedUser() void Public	Metoda pozada bussiness logiku aby upravila prave prihlaseneho uzivatele.	
recognizeLogin() String Public	Metoda urci o jakou instanci uzivatele se aktualne jedna, zda-li je prihlasen administrator, student nebo ucitel. @return loginStudent - je prihlasen student @return loginKantro je prihlasen kantor @return loginAdmin je prihlaen admin	
resetPassoword() void Public	Metoda resetuje heslo uzivateli a nastavi mu defaultni heslo	
setLoggedPerson() void Public		**Uzivatel** [in] **loggedPerson**

testBean

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	backingBean *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{D2E8CD39-0ADE-498b-9DAD-9ABE5C9615C7}

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "testBean")@RequestScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public testBean	Package app ApplicationLocal	
Association Source -> Destination	Public testBean	Package ukolD IUkolD	
Association Source -> Destination	Public testBean	Package uzivD IUzivatelD	
Association Source -> Destination	Public testBean	Package zadaD IZadaniD	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
something String Package		*Default:*
ukolD IUkolD Package		*Default:*
uzivD IUzivatelD Package		*Default:*
zadaD IZadaniD Package		*Default:*

Operations

Method	Notes	Parameters
callPersist() void Public		
getSomething() String Public		
init() void Public		

setSomething() void Public		String [in] something
testBean() Public	Creates a new instance of testBean	

groupBean

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: PL
Detail: Created on 23.11.2011. Last modified on 23.11.2011
GUID: {E06B9678-EBDE-45ea-BF1F-62EF3D40EAB7}

groupBean - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
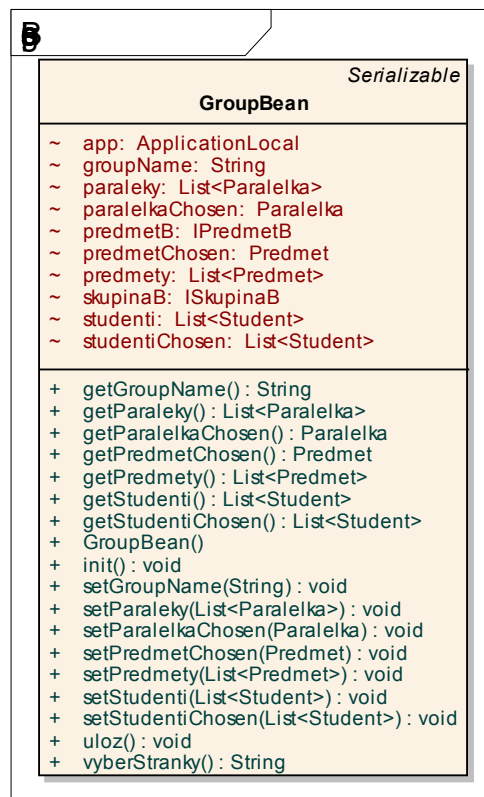*GUID:*   {EA526702-761B-4370-9BD5-1A46E9BC6E92}

Figure: 12

GroupBean

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	groupBean *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{F3A2DCA9-265D-4e0f-BA90-7CA08798CED1}

Beana, kterĂˇ si slouĹľĂ pro pĹ™idĂˇvĂˇnĂ studentu do skupin

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "groupBean")@SessionScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public GroupBean	Package app ApplicationLocal	
Association Source -> Destination	Public GroupBean	Package paralelkaChosen Paralelka	
Association Source -> Destination	Public GroupBean	Package predmetB IPredmetB	
Association Source -> Destination	Public GroupBean	Package predmetChosen Predmet	
Association Source -> Destination	Public GroupBean	Package skupinaB ISkupinaB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]

groupName String Package		*Default:*
paraleky List<Paralelka> Package		*Default:*
paralelkaChosen Paralelka Package		*Default:*
predmetB IPredmetB Package		*Default:*
predmetChosen Predmet Package		*Default:*
predmety List<Predmet> Package		*Default:*

skupinaB ISkupinaB Package		*Default:*
studenti List<Student> Package		*Default:*
studentiChosen List<Student> Package		*Default:*

Operations

Method	Notes	Parameters
getGroupName() String Public		
getParaleky() List<Paralelka> Public		
getParalelkaChosen() Paralelka Public		
getPredmetChosen() Predmet Public		
getPredmety() List<Predmet> Public		
getStudenti() List<Student> Public		
getStudentiChosen() List<Student> Public		
GroupBean() Public		
init() void Public		
setGroupName() void Public		**String** [in] **groupName**

setParaleky() void Public		**List<Paralelka>** [in] **paraleky**
setParalelkaChosen() void Public		**Paralelka** [in] **paralelkaChosen**
setPredmetChosen() void Public		**Predmet** [in] **predmetChosen**
setPredmety() void Public		**List<Predmet>** [in] **predmety**
setStudenti() void Public		**List<Student>** [in] **studenti**
setStudentiChosen() void Public		**List<Student>** [in] **studentiChosen**
uloz() void Public	UloĹľĂ data o nove skupine do databĂˇze	
vyberStranky() String Public	Funkce, kterĂˇ vracĂ odkaz na stranku, na kterou budeme kterĂˇ se ma naĂŤĂst. Nejtrve se naĂŤĂta strĂˇnka s vĂ˝berem pĹ™edmĂ›tu, nĂˇslednĂ› strĂˇnka pro vĂ˝bÄ›r paralelky a potĂ© strĂˇnka pro vĂ˝bÄ›r studentĹŽ do skupiny. @return	

subjectBean

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: PL
Detail: *Created on 23.11.2011. Last modified on 23.11.2011*
GUID: {54A6550B-759E-4d25-AE25-5089AAD34E54}

subjectBean - *(Logical diagram)*
Created By: papa *on* 23.11.2011
Last Modified: 23.11.2011
Version: 1.0. *Locked:* False
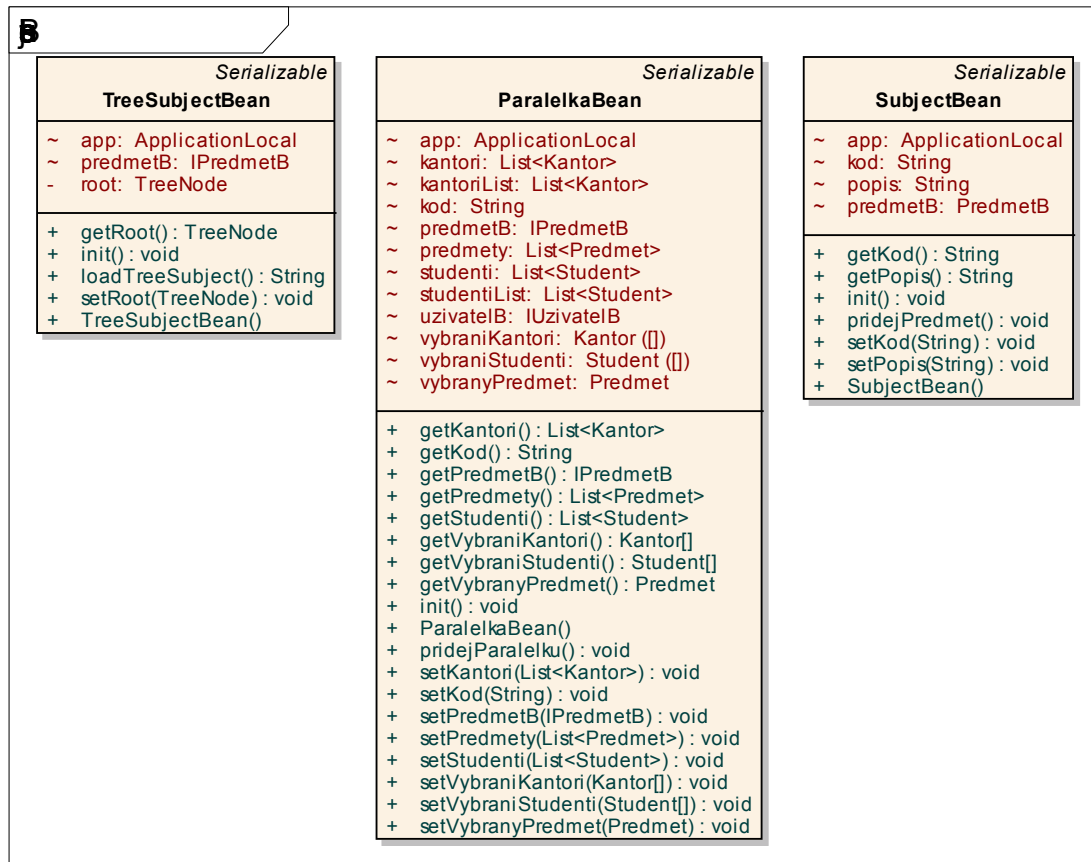*GUID:*   {75E0887C-2871-4bd4-8F1C-330C1DB5F69F}

Figure: 13

ParalelkaBean

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	subjectBean *Keywords:*
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011.*
GUID:	{6317EDA4-90FF-4401-8695-2EAD929F8449}

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "paralelkaBean")@RequestScoped.

Connections

Connector	Source	Target	Notes

Association Source -> Destination	Public ParalelkaBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public ParalelkaBean	Package vybraniKantori Kantor	
Association Source -> Destination	Public ParalelkaBean	Package vybraniStudenti Student	
Association Source -> Destination	Public ParalelkaBean	Package vybranyPredmet Predmet	
Association Source -> Destination	Public ParalelkaBean	Package app ApplicationLocal	
Association Source -> Destination	Public ParalelkaBean	Package predmetB IPredmetB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
kantori List<Kantor> Package		*Default:*
kantoriList List<Kantor> Package		*Default:*

kod String Package		*Default:*
predmetB IPredmetB Package		*Default:*
predmety List<Predmet> Package		*Default:*
studenti List<Student> Package		*Default:*
studentiList List<Student> Package		*Default:*
uzivatelB IUzivatelB Package		*Default:*

vybraniKantori Kantor Package Collection		*Default:*
vybraniStudenti Student Package Collection		*Default:*
vybranyPredmet Predmet Package		*Default:*

Operations

Method	Notes	Parameters
getKantori() List<Kantor> Public		
getKod() String Public		
getPredmetB() IPredmetB Public		
getPredmety() List<Predmet> Public		
getStudenti() List<Student> Public		
getVybraniKantori() Kantor Public		
getVybraniStudenti() Student Public		
getVybranyPredmet() Predmet Public		
init() void Public		
ParalelkaBean() Public		
pridejParalelku() void Public	Funkce si vybere data z .jsf stǎˇnky (vybranǍ studenti, vybranǍ kantoĹ™i, kǍłd paralelky a	

	pĹ™edmet). VybranĂˇ data uloĹľĂ˝ do databĂˇze	
setKantori() void Public		**List<Kantor>** [in] **kantori**
setKod() void Public		**String** [in] **kod**
setPredmetB() void Public		**IPredmetB** [in] **predmetB**
setPredmety() void Public		**List<Predmet>** [in] **predmety**
setStudenti() void Public		**List<Student>** [in] **studenti**
setVybraniKantori() void Public		**Kantor[]** [in] **vybraniKantori**
setVybraniStudenti() void Public		**Student[]** [in] **vybraniStudenti**
setVybranyPredmet() void Public		**Predmet** [in] **vybranyPredmet**

SubjectBean

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: subjectBean   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {A466891A-5E6F-4e6b-BE08-08BAE442E255}

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "subjectBean")@RequestScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public SubjectBean	Package app ApplicationLocal	

Association Source -> Destination	Public SubjectBean	Package predmetB PredmetB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
kod String Package		*Default:*
popis String Package		*Default:*
predmetB PredmetB Package		*Default:*

Operations

Method	Notes	Parameters
getKod() String Public		
getPopis() String Public		
init() void Public		
pridejPredmet() void Public	funkce volajici funkci ... z bussiness vrstvy pro pridani do databaze	
setKod() void		**String** [in] **kod**

Public		
setPopis() void Public		**String** [in] **popis**
SubjectBean() Public		

TreeSubjectBean

Type: **Class   Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: subjectBean   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {5420EDBE-2886-4d89-A58C-43E0A9C6A1ED}

Custom Properties

• isActive = False

Tagged Values

• annotations = @ManagedBean(name = "treeSubjectBean")@RequestScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public TreeSubjectBean	Package predmetB IPredmetB	
Association Source -> Destination	Public TreeSubjectBean	Package app ApplicationLocal	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]

predmetB IPredmetB Package		Default:
root TreeNode Private		Default:

Operations

Method	Notes	Parameters
getRoot() TreeNode Public		
init() void Public		
loadTreeSubject() String Public	Funkce nejprv naĂŤte data z databĂˇze a potĂ© data pĹ™iĹ™adĂ do jednotlivĂ˝ch ĂşrovnĂ. @return String se jmenem svĂ© strĂˇnky.	
setRoot() void Public		TreeNode [in] root
TreeSubjectBean() Public		

taskBeans

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	PL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{0E7B3872-EEAB-430c-B1F1-53551812D103}

taskBeans - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{6C82E428-97D6-439f-902A-BAA36ADC139D}

Figure: 14

AssignTask

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: taskBeans   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {B304D86A-303C-4aee-9DFD-BA5730328802}

Trida umoznuje pridani zadani jednotlivym studentum ci mnozine techto studentu

Custom Properties

- isActive = False

- annotations = @ManagedBean(name = "assignTask")@SessionScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AssignTask	Package app ApplicationLocal	
Association Source -> Destination	Public AssignTask	Package currChosen Student	
Association Source -> Destination	Public AssignTask	Package kantrosSubjectToFilter Predmet	
Association Source -> Destination	Public AssignTask	Package loggedKantor Kantor	
Association Source -> Destination	Public AssignTask	Package paralelChosen Paralelka	
Association Source -> Destination	Public AssignTask	Package paralelkaAsSubject Paralelka	
Association Source -> Destination	Public AssignTask	Package subjectB PredmetB	
Association Source -> Destination	Public AssignTask	Package taskB UkolB	
Association Source -> Destination	Public AssignTask	Package taskChosen Zadani	
Association Source -> Destination	Public AssignTask	Package temp Paralelka	
Association Source -> Destination	Public AssignTask	Package userB UzivatelB	

Attributes

Attribute	Notes	Constraints and tags

app ApplicationLocal Package		*Default:* [annotations = @EJB]
currChosen Student Package Collection		*Default:*
filtredParalelBySubject List<Paralelka> Package		*Default:*
finalTaskList List<Ukol> Package		*Default:*
kantorParalel List<Paralelka> Package		*Default:*
kantorsTask List<Zadani> Package		*Default:*

kantorSubjects List<Predmet> Package		*Default:*
kantrosSubjectToFilter Predmet Package Collection		*Default:*
loggedKantor Kantor Package		*Default:*
paralelChosen Paralelka Package Collection		*Default:*
paralelkaAsSubject Paralelka Package Collection		*Default:*
studentsChosen List<Student> Package		*Default:*

subjectB PredmetB Package		*Default:*
subjectChosen List<Predmet> Package		*Default:*
taskB UkolB Package		*Default:*
taskChosen Zadani Package Collection		*Default:*
temp Paralelka Package		*Default:*
userB UzivatelB Package		*Default:*

Operations

Method	Notes	Parameters
assigenTask() String Public	Metoda priradi ukol, ci ukoly skupine studentu ci samotnemu studentovi.	

confirmTask() String Public	Metoda pozada bussiness vrstu o vytvoreni ukolu pro aktualne vybrane studenty, pro aktualne vybrany seznam ukolu a pro ucitele, ktery tento ukol zadal.	
filterByParalel() void Public	Metoda vyfiltruje studenty na prave zvolene paralelce ci skupine paralelek, metoda zajisti odstraneni zobrazeni duplicitnich zazanum.	
filterBySubject() void Public	Metoda vyfiltruje paralelky a studenty na aktualne zvolenem predmetu ci skupine predmetu metoda zajisti odstraneni zobrazeni duplicitnich zazanmu	
getCurrChosen() Student Public		
getFiltredParalelBySubject() List<Paralelka> Public		
getFinalTaskList() List<Ukol> Public		
getKantorParalel() List<Paralelka> Public		
getKantorsTask() List<Zadani> Public		
getKantorSubjects() List<Predmet> Public		
getKantrosSubjectToFilter() Predmet Public		
getParalelChosen() Paralelka Public		
getParalelkaAsSubject() Paralelka Public		
getPrincipal() Principal Private	Metoda slouzi k vraceni aktualne prihlaseneho uzivatele (Principala), JSF JAAS framework @return aktualne prihlaseneho Principala	
getStudentsChosen() List<Student> Public		
getSubjectChosen() List<Predmet> Public		
getTaskChosen() Zadani Public		
getTemp() Paralelka Public		
init() void		

Public		
loadClass() String Public		
setCurrChosen() void Public		**Student[]** [in] **currChosen**
setFiltredParalelBySubje ct() void Public		**List<Paralelka>** [in] **filtredParalelBySubject**
setFinalTaskList() void Public		**List<Ukol>** [in] **finalTaskList**
setKantorParalel() void Public		**List<Paralelka>** [in] **kantorParalel**
setKantorsTask() void Public		**List<Zadani>** [in] **kantorsTask**
setKantorSubjects() void Public		**List<Predmet>** [in] **kantorSubjects**
setKantrosSubjectToFilte r() void Public		**Predmet[]** [in] **kantrosSubjectToFilter**
setParalelChosen() void Public		**Paralelka[]** [in] **paralelChosen**
setParalelkaAsSubject() void Public		**Paralelka[]** [in] **paralelkaAsSubject**
setStudentsChosen() void Public		**List<Student>** [in] **studentsChosen**
setSubjectChosen() void Public		**List<Predmet>** [in] **subjectChosen**
setTaskChosen() void Public		**Zadani[]** [in] **taskChosen**
setTemp() void Public		**Paralelka** [in] **temp**

studentTask

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: taskBeans   *Keywords:*

Trida slouzi k podpore view studenta, umi zobrazit jeho ukoly, jeho odevzdane ukoly, prirazene ukoly.

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "studentTask")@RequestScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public studentTask	Package student Student	
Association Source -> Destination	Public studentTask	Package userB UzivatelB	
Association Source -> Destination	Public studentTask	Package ukolB UkolB	
Association Source -> Destination	Public studentTask	Package app ApplicationLocal	
Association Source -> Destination	Public studentTask	Package loggedKantor Kantor	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
loggedKantor Kantor Package		*Default:*

setOfTasks List<Ukol> Package		*Default:*
student Student Package		*Default:*
ukolB UkolB Package		*Default:*
userB UzivatelB Package		*Default:*

Operations

Method	Notes	Parameters
getPrincipal() Principal Private	Metoda slouzi k vraceni aktualne prihlaseneho uzivatele (Principala), JSF JAAS framework @return aktualne prihlaseneho Principala	
getSetOfTasks() List<Ukol> Public		
init() void Public		
loadAssignedTasks() String Public	Metoda pozada bussiness logiku o vraceni aktualne zadanych ukolu, na zaklade studenta, ktery je prihlasen. @return assignedTask - smerovaci pravidlo	
setSetOfTasks() void Public		**List<Ukol>** [in] **setOfTasks**
taskToStudent() void Public		

taskBean

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: taskBeans   *Keywords:*
Detail: Created on 23.11.2011. Last modified on 23.11.2011.
GUID: {0DAC596A-5BE4-45c7-BEFD-7AADF641D79C}

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "taskBean")@SessionScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public taskBean	Package zadaniBus ZadaniB	
Association Source -> Destination	Public taskBean	Package app ApplicationLocal	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
deadline Date Package		*Default:*

taskDescription String Package		*Default:*
taskName String Package		*Default:*
tasks List<Zadani> Package		*Default:*
zadaniBus ZadaniB Package		*Default:*

Operations

Method	Notes	Parameters
createTask() void Public	Metoda preda business logice parametry, s kterymi se ma vytvorit nove zadani.	
getDeadline() Date Public		
getPrincipal() Principal Private	Metoda slouzi k vraceni aktualne prihlaseneho uzivatele (Principala), JSF JAAS framework @return aktualne prihlaseneho Principala	
getTaskDescription() String Public		
getTaskName() String Public		
getTasks() List<Zadani> Public		
init() void Public		
loadAllTasks() String	Metoda pozada business logiku o seznam vsech	

Public	zadani, ktere vytvoril prave prihlaseny uzivatel (kantor) @return edittaskKantr - navigaÄŤnĂ pravidlo	
modifyTask() String Public		
setDeadline() void Public		**Date** [in] **deadline**
setTaskDescription() void Public		**String** [in] **taskDescription**
setTaskName() void Public		**String** [in] **taskName**
setTasks() void Public		**List<Zadani>** [in] **tasks**

userBean

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	PL
Detail:	*Created on 23.11.2011. Last modified on 23.11.2011*
GUID:	{5851F734-17A9-49e4-8699-D627842DFD9C}

userBean - *(Logical diagram)*

Created By:	papa *on* 23.11.2011
Last Modified:	23.11.2011
Version:	1.0. *Locked:* False
GUID:	{CC9CFE91-40FE-4a3f-89EF-C079893582FA}

Figure: 15

CheckStudentBean

Type:	**Class**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	userBean *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{960E3D6D-56F5-4508-9B85-9F6B78824AF4}

Beana souĹĺˇĂcĂ k vĂˇpisu a mazĂˇnĂ studentĹŻ.

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "checkStudentBean")@RequestScoped.

Connections

Connector	Source	Target	Notes

Association	Public	Package app	
Source -> Destination	CheckStudentBean	ApplicationLocal	
Association	Public	Package chosen	
Source -> Destination	CheckStudentBean	Uzivatel	
Association	Public	Package uzivatelB	
Source -> Destination	CheckStudentBean	IUzivatelB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
chosen Uzivatel Package		*Default:*
studenti List<Student> Package		*Default:*
uzivatelB IUzivatelB Package		*Default:*

Operations

Method	Notes	Parameters
CheckStudentBean() Public		
deleteStudent() String Public	Metoda pozada business logiku o odstraneni aktualniho vybraneho studenta a aktalizuje seznam studentu v DB	

	@return checkstudentAdmin - navigacni pravidlo	
getChosen() Uzivatel Public		
getPrincipal() Principal Private	Metoda slouzi k vraceni aktualne prihlaseneho uzivatele (Principala), JSF JAAS framework @return aktualne prihlaseneho Principala	
getStudenti() List<Student> Public		
init() void Public		
setChosen() void Public		**Uzivatel** [in] **chosen**
setStudenti() void Public		**List<Student>** [in] **studenti**

FileBean

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: userBean   *Keywords:*
Detail: *Created on 23.11.2011. Last modified on 23.11.2011.*
GUID: {249A6E2C-6B74-4ca5-8F71-F787AEC1D99B}

Trida se zameruje na nacteni souboru a predani je do business logiky.

Custom Properties
- isActive = False

Tagged Values
- annotations = @ManagedBean(name = "fileBean").

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public FileBean	Package app ApplicationLocal	
Association Source -> Destination	Public FileBean	Package uzivatelB IUzivatelB	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
img UploadedFile Package		*Default:*
uzivatelB IUzivatelB Package	instance v bussiness vrstve starajici se o uzivatele	*Default:*

Operations

Method	Notes	Parameters
handleFileUpload() void Public	Metoda nacte aktualni soubor a odesle ho do business logiky ke zpracovani.	**FileUploadEvent** [in] **event**
init() void Public		

UserBean

Type:	**Class Serializable**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	userBean *Keywords:*
Detail:	Created on 23.11.2011. Last modified on 23.11.2011.
GUID:	{5FAA3B51-F9D4-4a0e-8CFE-4F4E0BDBB535}

Beana pro pĹ™idÄˇvÄˇnÄ novÄ˝ch studenĹŻ.

Custom Properties

- isActive = False

Tagged Values

- annotations = @ManagedBean(name = "userBean")@RequestScoped.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public UserBean	Package uzivatelB IUzivatelB	
Association Source -> Destination	Public UserBean	Package app ApplicationLocal	

Attributes

Attribute	Notes	Constraints and tags
app ApplicationLocal Package		*Default:* [annotations = @EJB]
cssErr String Package		*Default:*
email String Package		*Default:*
err String Package	urĂŤeno pro vypis erroru a pro nastaveni css pro error zpravu	*Default:*

heslo String Package		*Default:*
id int Package		*Default:*
jmeno String Package		*Default:*
login String Package		*Default:*
prijmeni String Package		*Default:*
status String Package		*Default:*

uzivatelB IUzivatelB Package	instance v bussiness vrstve starajici se o uzivatele	*Default:*

Operations

Method	Notes	Parameters
getCssErr() String Public		
getEmail() String Public		
getErr() String Public		
getHeslo() String Public		
getJmeno() String Public		
getLogin() String Public		
getPrijmeni() String Public		
getStatus() String Public		
init() void Public		
nastalaChyba() void Private	nastavenÄ zpravy a css stylu pri chybÄ›	
pridejUzivatele() void Public	prida uĹľivatele do databaze, zavola funkci s bussiness vrstve addUzivatel	
setCssErr() void Public		**String** [in] **cssErr**
setEmail() void Public		**String** [in] **email**
setErr() void Public		**String** [in] **err**
setHeslo() void Public		**String** [in] **heslo**
setJmeno() void Public		**String** [in] **jmeno**
setLogin() void Public		**String** [in] **login**
setPrijmeni() void Public		**String** [in] **prijmeni**

setStatus() void Public		**String** [in] **status**
UserBean() Public		